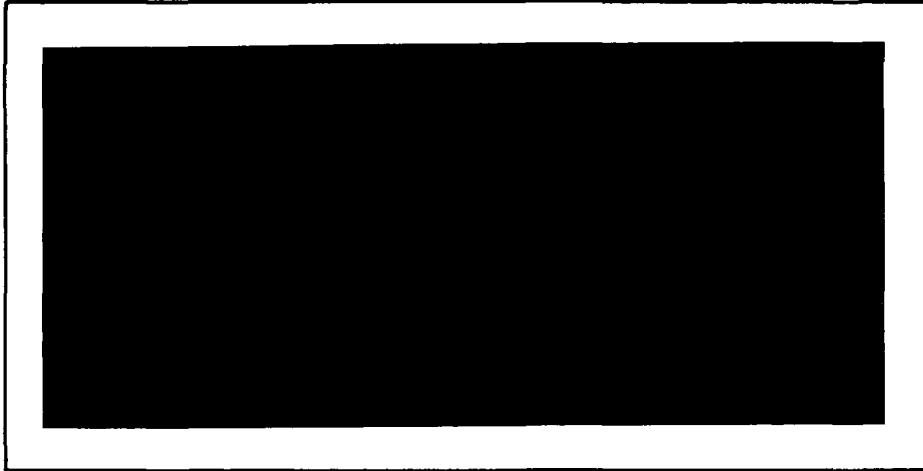


N73-32551



SCIENCE
Applications
INCORPORATED

CASE FILE
COPY

METHODS OF TREATING COMPLEX
SPACE VEHICLE GEOMETRY FOR
CHARGED PARTICLE RADIATION
TRANSPORT

SAI-74-549-HU

Final Report

Prepared by

C. W. Hill

for

George C. Marshall Space Flight Center

Huntsville, Alabama 35812

George C. Marshall Space Flight Center

Contract NAS8-28177

August 1973

SCIENCE APPLICATIONS, INCORPORATED

Suite 700, 2109 W. Clinton Ave., Huntsville, Alabama (205) 539-0001

	TABLE OF CONTENTS	PAGE
1.0	INTRODUCTION	1
2.0	BASIC GEOMETRY CONSIDERATIONS	4
2.1	SURFACES	4
2.2	VOLUME ELEMENTS	7
2.3	GEOMETRY DENSITY	12
2.4	EMBEDDING	13
2.5	TRANSFORMATIONS	18
3.0	GEOMETRY MODEL SCANNING	23
3.1	RAY GENERATION	23
3.2	RAY TRACING	29
3.3	EFFICIENCY SCHEMES FOR DEFENSE GEOMETRY RAY TRACING	35
3.4	EFFICIENCY SCHEMES FOR SPARSE GEOMETRY RAY TRACING	36
4.0	DATA FORMAT	50
4.1	SIGMA DATA FORMAT	50
4.2	MEVDP DATA FORMAT	51
4.3	HSV DATA FORMAT	53
5.0	DATA CHECKING	55
5.1	SIGMA DATA CHECKING	55
5.2	MEVDP DATA CHECKING	56
5.3	HSV DATA CHECKING	60

		PAGE
6.0	CODE COMPARISON AND EVALUATION	65
6.1	SIGMA GEOMETRY CAPABILITY	65
6.2	MEVDP AND HSV GEOMETRY CAPABILITY	67
6.3	CODE SPEED COMPARISONS	69
6.4	EVALUATION SUMMARY	74
7.0	RECOMMENDATIONS FOR FUTURE GEOMETRY DEVELOPMENTS	76
7.1	SUGGESTED SIGMA IMPROVEMENTS	76
7.2	SUGGESTED MEVDP IMPROVEMENTS	79
7.3	SUGGESTED HSV IMPROVEMENTS	82

LIST OF TABLES AND FIGURES

		<u>Page</u>
TABLE 6.1	Relative Speed Of Geometry Codes	71
FIGURE 1	Simple Volume Element	9
FIGURE 2	Volume Element With Reentrant Boundaries	11
FIGURE 3	MEDVP Volume Elements	15
FIGURE 4	Insulated Tank	17
FIGURE 5	Representation of Ray Tracing	24
FIGURE 6	Ray Spherical Coordinates	28
FIGURE 7	Polyhedral Envelope for HSV ¹⁰	30
FIGURE 8	Dosimeter (DSCS) and Absolute (ABCS) Coordinate Systems for MEVDP ¹⁹	40
FIGURE 9	Rotated Detector Coordinate System (RDCS) for MEVDP ¹¹	42
FIGURE 10	Projection of Cylinder (Cone) Base Plane for MEVDP ¹⁹	43
FIGURE 11	Projection of Hexahedron Face for MEVDP ¹⁹	45
FIGURE 12	Schematic of HSV Super Region Nesting	47
FIGURE 13	Sigma-Drawn Section Through Skylab Geometric Model ²⁵	57
FIGURE 14	MEVDP Computer Drawing of the Eye ¹⁷	59
FIGURE 15	HSV Computer Drawing of the Eye ¹⁷	63
FIGURE 16	MEVDP Hexahedron Point Ordering	80

ABSTRACT

This report reviews current methods of treating complex geometry models for space radiation transport calculations. The geometric techniques used in three computer codes are outlined. Evaluations of geometric capability and speed are provided for these codes. Although no code development work is included in this effort, several suggestions for significantly improving complex geometry codes are offered.

FOREWARD

This report completes a three part survey of space radiation shielding techniques sponsored by the National Aeronautics and Space Administration. Proton and alpha transport are discussed in an Oak Ridge National Laboratory report⁽²⁾ written by Alsmiller et al. Electron and bremsstrahlung transport are discussed in a Science Applications report⁽²³⁾ written by Shreve and Lonergan. The present report is intended to cover the current state-of-the-art in treating complex geometry models for space radiation transport calculations.

The Contract Monitor is Martin O. Burrell of the George C. Marshall Space Flight Center, NASA. His guidance and patience are appreciated.

The services of Mrs. Betty F. Maskewitz and her co-workers at the Radiation Shielding Information Center were invaluable to this effort. The RSIC is much more than a clearinghouse for codes and bibliographies; it is a Center where help is available and freely offered to the radiation analyst.

Real assistance from friendly competitors is unexpected treasure. Bob Langley and Pres Billings of McDonnell Douglas, and Bob Liley of North American Rockwell have been generous with their time and data. Their explanations and opinions have been incorporated into this report in several places, though they should be held blameless for any misrepresentation contained herein. The suggestions for improving their codes, presented in Section 7, are made without consultation due to shortage of time.

This report is written for the user and developer of complex geometry codes rather than for the presentation of data. It should be understood that the writer of this report is most familiar with his own work and consequently a bias is necessarily present.

1.0 INTRODUCTION

The study of energetic charged particle transport through complex configurations has been intensive over the last decade. Until the 1960's, charged particle transport studies were driven by interest in cosmic radiation, accelerators, and the explication of nuclear forces. Calculations were usually made for slab or infinite media geometries except for multiple scatter collimator corrections. With the advent of the space program, questions concerning the hazards of space radiations to man and equipment within space vehicles became critical. As the understanding of radiation sources and transport grew, the need to represent space vehicles with more realistic geometry models increased.

From 1959 to 1963, the portion of the aerospace community concerned with charged particle transport focused on the transmission of primary and secondary radiation through simple geometries. Current reviews of knowledge in this area have recently been made by Alsmiller et al.⁽²⁾ at Oak Ridge for protons and alpha particles, and by Shreve and Lonergan⁽²³⁾ at Science Applications for electrons and bremsstrahlung.

By 1963, several groups started thinking in terms of complex geometry codes because of a number of factors. Spacecraft could carry little dead weight in the form of passive shielding, so the shielding afforded by structure and equipment should be taken into account. The configurations of spacecraft do not lend themselves to simple geometry representations. Several components of space radiations exhibit steep dose-versus-thickness curves. This feature means faithful geometry models are necessary for computational

accuracy. Finally, computer size and speed increases made detailed geometry treatment feasible. For these reasons, at least seven groups started complex geometry shielding codes in the period from 1963 to 1965. Codes started then but not widely used today include:

- K019 - by Malone at the Manned Spacecraft Center,
NASA, Houston,⁽²¹⁾
- CARS - by Fortney at Northrop,⁽⁷⁾
- Unnamed - by Madey at Republic,⁽²⁰⁾
- Unnamed - by Bresticker at Grumman.⁽⁴⁾

Three complex geometry codes for treating space radiation transport are currently in use, SIGMA, MEVDP, and HSV. SIGMA⁽¹³⁾ was developed at Douglas (now McDonnell Douglas) by Jordan. Langley and Billings are now caretakers for the SIGMA code. They have expanded the code's capabilities several times. Some features of the latest version were communicated privately just prior to publication of this report and are outlined briefly in a later section.

MEVDP⁽¹⁹⁾ (Modified Elemental Volume Dose Program) was developed by Liley and Hamilton at North American (now North American Rockwell). An early version was called EVDP. The geometric shapes used are similar to those in K019. Some features of the latest version were communicated privately just prior to publication of this report and are outlined in a later section.

HSV^(8, 9, 10, 12) (Huntsville Space Vehicle code) was developed by Hill, Simpson, Douglass, and others at Lockheed Aircraft. Hill at Science Applications is caretaker of the code. The first version, LSVDC, was modified appreciably, converted to FORTRAN IV, and

renamed LSVDC4. Later, the geometry and dose sections were modified and the code was renamed HSV.

This report deals primarily with geometry considerations. Dose computation techniques are detailed in user manuals. Briefly, SIGMA computes proton dose by table look-up in data tables prepared by the CHARGE physics code⁽²⁵⁾. MEVDP computes proton dose using an approximation valid from 5 to 200 MeV. Documentation is not currently available for MEVDP electron dose methods. HSV computes proton and alpha dose by a method due to Burrell⁽⁵⁾ with an approximation valid from 5 to 1000 MeV/nucleon. HSV computes electron and bremsstrahlung doses by methods outlined in the user manual, and radiation fogging effects for photographic film by methods partially documented⁽¹¹⁾.

Section 2 of this report discusses basic geometry considerations concerning surfaces and volume elements. Section 3 discusses methods of scanning or ray tracing geometry models. Section 4 discusses data formats for the three codes in active use. Section 5 discusses the critical problem of data checking. Section 6 compares and evaluates present codes. Section 7 contains recommendations for improving present codes.

2.0 BASIC GEOMETRY CONSIDERATIONS

This section outlines the fundamental aspects of geometry codes. The basic element treated is the surface. Surfaces may be combined in various ways to form the building blocks - called volume elements in this report - of a geometry model. Section 2.3 defines two methods of combining volume elements into a geometry model; dense packing or sparse packing. SIGMA follows the dense packing rules while MEVDP and HSV follow the sparse packing rules.

Section 2.4 discusses the embedding or overlap of volume elements. This feature is currently used only in the sparse packing type of code. One recommendation of this report is that the embedding feature be added to SIGMA in order to increase its geometry capabilities while retaining its speed advantage.

Section 2.5 discusses transformations which may be used to move groups of volume elements in sparse geometry models. This feature is not applicable to dense geometry codes.

2.1 SURFACES

The basic element of the geometric representations described here is the surface. The first degree surface is the plane; algebraically it is described by Equation 1.

$$AX + BY + CZ + D = 0 \quad (1)$$

The quantities X, Y, and Z are variables in three-space; A, B, C, and D are constants for a particular plane. The value of at least one of the coefficients (A, B, C) must be non-zero. If the value of D is

zero, the origin of the coordinate system is included in the plane.

A plane surface divides all space into two parts. Thus, a plane is said to have polarity. The polarity of a plane may be defined with the aid of Equation 2. If the XYZ coordinates of a point on the plane

$$AX + BY + CZ + D = q \quad (2)$$

are inserted into Equation 2, the value of q is zero. All points on one side of the plane yield positive values of q ; all points on the other side, negative. The polarity of the plane may be reversed by changing the sign of all coefficients A , B , C , and D . The polarity is used to define on which side of a surface a volume element (defined in Section 2.2) is located.

Second degree surfaces are represented by Equation 3. The usual terminology is to label these quadric surfaces. Again, the polarity

$$\begin{aligned} &AX^2 + BY^2 + CZ^2 + DXY + EXZ + FYZ + GX + HY \\ &+ PZ + Q = 0 \end{aligned} \quad (3)$$

of the surface may be defined by replacing the zero on the right side by q . The value of q will be zero for points on the surface, positive on one side, and negative on the other. The polarity may be reversed by changing the sign of all coefficients A through Q .

Quadric surfaces are divided into the categories listed below.

- ellipsoid
- cylinder
- elliptic cone
- elliptic hyperboloid of one sheet
- elliptic hyperboloid of two sheets
- elliptic paraboloid
- hyperbolic paraboloid

It should be understood that the ellipsoid contains as degenerate cases; the prolate spheroid, the oblate spheroid, and the sphere. The cylinder includes three special cases; elliptic, parabolic, and hyperbolic cylinders. The elliptic cylinder includes, as a degenerate case, circular cylinders. The other four elliptic surfaces contain circular surfaces as degenerate cases.

Note that regions of a space divided by a quadric surface need not be connected to have the same polarity. The regions within both nappes of a cone possess the same polarity even though they meet at a point only. Two separated regions with the same polarity exist for the hyperboloid of two sheets.

The toroid is a fourth degree or quartic surface that is sometimes used in geometry calculations. The general quartic contains 34 terms. The canonical form for the toroid is given as Equation 4. The polarity of this surface may be switched by changing the sign of A, B, and D.

$$AZ^2 + B [(X^2 + Y^2)^{1/2} - C]^2 + D = 0 \quad (4)$$

2.2 VOLUME ELEMENTS

The surfaces described may be combined in various ways to form volume elements. Volume elements - which may also be termed elemental volumes, regions, zones, or shields - are the basic building blocks in constructing a representation of a complex configuration. A variety of options are available for restricting the form of volume elements; the options chosen tend to define the "flavor" of a program and the pattern used to specify geometric data. The discussion which follows will first point out characteristics common to most geometry programs, and will then illustrate a few of the choices which may be made for convenience, simplicity, or flexibility.

Four characteristics of volume elements are common to most geometry treatments; boundedness, material identification, homogeneity, and unambiguous polarity.

The first characteristic, boundedness, means that volume elements are finite in extent. Whereas cylinders and plane slabs may extend to infinity in simple geometry programs, this behavior is rare in complex geometry programs. Usually such volume elements are truncated to a finite region of space. Some programs use an external void; this term refers to a boundary beyond which no volume elements are defined.

To specify the second characteristic, a parameter is associated with each volume element to label the material or mixture of materials composing it. Voids may have their own material identification number, or may be specified with a zero-density.

The third characteristic is homogeneity. The interior of a volume element is usually homogeneous material or void. An exception

to this rule arises where one volume element is embedded within another. This approach will be discussed in Section 2.4. Another exception is often treated in cosmic ray transport where an exponential atmosphere may be modeled. A third exception which may be called variable thickness has been discussed over the last ten years but has probably not yet been incorporated into a geometry program. As an example of variable thickness, consider an electronics box containing several parallel circuit boards. Neglecting lumped components, a ray passing through the box perpendicular to the circuit boards penetrates a fixed mass thickness regardless of the point of entry. A ray passing through the box parallel to the circuit boards may penetrate only the box walls, or may also penetrate varying amounts of material depending on the entry point and direction of travel. This case could be treated by modeling the electronic box in detail with many volume elements, or as a simple box with a functional mass thickness dependent on the entry point and direction of a ray. Such a feature might also be used to simplify treatment of the skin of a space vehicle which is often strengthened by hundreds of lands or ridges.

The fourth volume element characteristic common to all programs is unambiguous polarity. This term means that every point within a volume element will have the same polarity with respect to all bounding surfaces. Further, every point outside the volume element will have opposite polarity with respect to at least one bounding surface. A simple volume element with polarity indicated is shown in Figure 1.

Aside from the above four volume element characteristics usually common to all complex geometry schemes, a number of choices

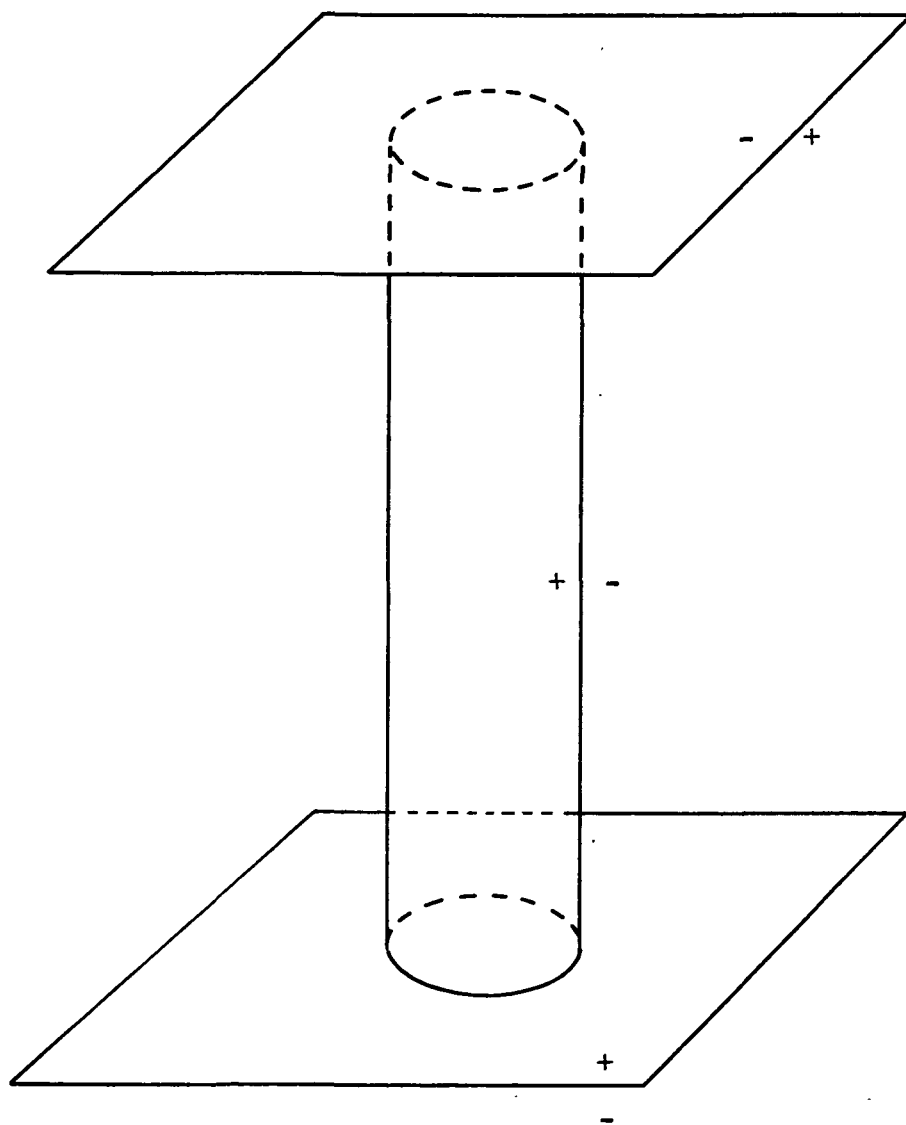


Figure 1. Simple Volume Element

are available in defining volume elements. One such choice is the type of surface permitted. Planes, circular cylinders, circular cones, and spheres are often included. Some geometry programs permit additional types of quadric surfaces up to the general quadric. Toroids, helical forms, and other surfaces may also be included. Figures of rotation may be restricted to have their principal axis parallel to a coordinate axis. This choice simplifies the algebra and shortens computational time at the expense of flexibility.

Another choice deals with the way surfaces may be combined to form volume elements. For example, figures bounded only by planes may be restricted to wedges (5 faces) and/or hexahedra (6 faces). Programs with such restrictions usually compensate for this limitation through the use of embedding which will be described in Section 2.4. Another example is restriction to ellipsoids, cylinders or one nappe of a cone, suitably truncated by the desired number of planes perpendicular to an axis of symmetry. These restrictions simplify program logic and data preparation.

Another choice, similar to the preceding one, deals with convexity of the volume element. In simple terms the surface of a convex volume element may be hit by a ray at two points (penetration), one point (tangent), or no points (miss). A grazing incidence is treated as a special case (most programs consider grazing to be a miss). More general programs do not require convexity provided the polarity criterion is met. Figure 2 shows a truncated cylinder with two "bites" taken out by spherical surfaces. Each sphere is called a re-entrant surface here. Ray 1 enters the cylinder exits and enters through the spherical surface, and finally leaves through the cylindrical surface.

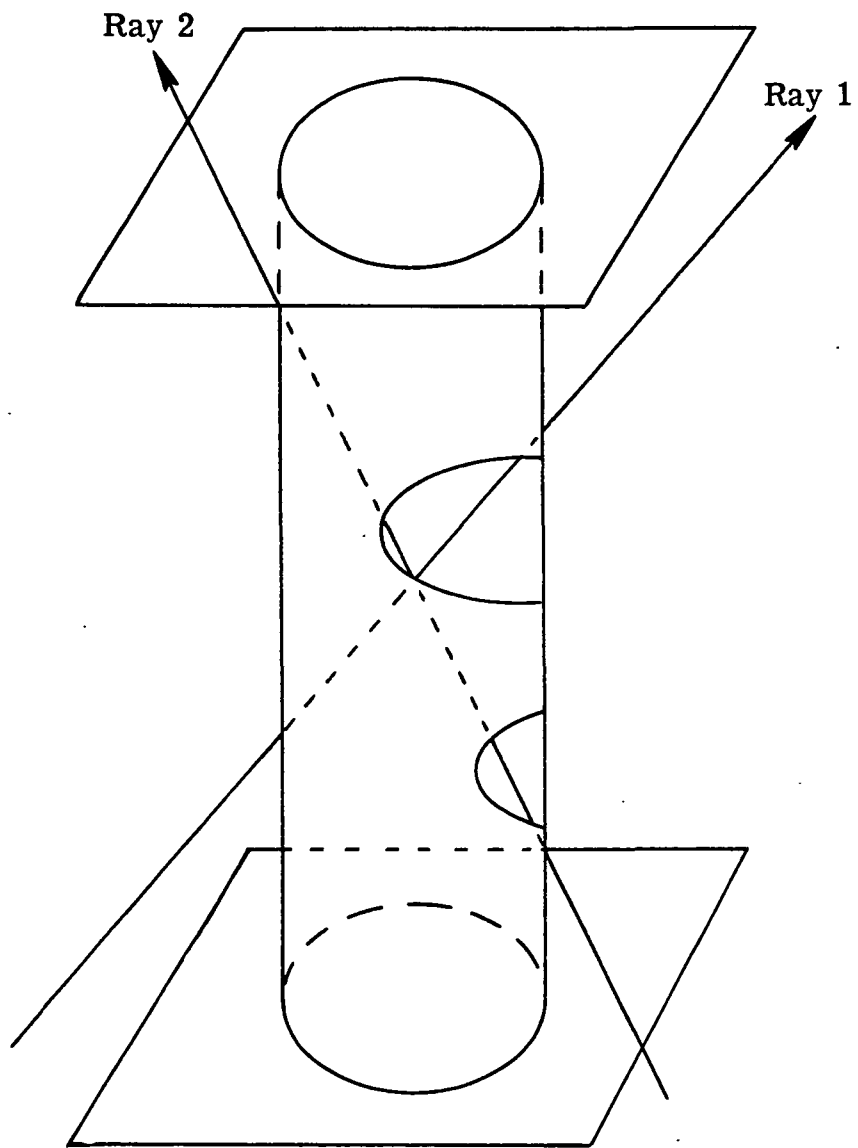


Figure 2. Volume Element With Reentrant Boundaries

Still, the volume element satisfies the polarity criterion and is valid unless convexity is required by the program logic.

A simple extension of the above argument shows that volume elements which are not simply connected may be permitted if desired. One sphere may be completely within the cylinder. If, in Figure 2, the "bite" is extended until the cylinder is completely severed by the sphere (or by several spheres), the single volume element will contain non-contiguous regions. Such disjoint volume elements are valid providing the polarity criterion is met and program provisions for these cases are made.

The special options outlined above pose serious choices for the analyst designing a geometry program. He may elect to go with simple convex shapes in order to simplify program logic. This choice also reduces learning time for the user. The extreme alternate - complex shapes without the convexity requirement - is more difficult to program and increases user learning time. The advantage of the latter choice lies in its capability of treating unusual shapes and reducing the number of volume elements required. An intermediate choice of complex, convex shapes is also available, but probably combines the disadvantages of both of the others without compensating gains.

2.3 GEOMETRY DENSITY

The terms "dense geometry" and "sparse geometry" are not widely used in the radiation transport community or elsewhere and will be defined here.

A dense geometry means that every point not on a surface in the configuration lies within a volume element; i. e., the volume

elements are packed together without gaps. The part of space beyond an encompassing boundary is labeled external void and tracking or ray tracing is terminated when that boundary is crossed. Rays which encounter an unintentional gap cause a "lost" signal to be issued and an error exit to be taken. The use of a dense geometry permits efficient ray tracing. Simple tables may be set up so that a routine first tests those volume elements most likely to be encountered after leaving a particular volume element. This type of geometry is almost always used in reactor and weapon radiation transport calculations.

A sparse geometry means that gaps may be left between volume elements. This feature leads to less efficient ray tracing and greater effort in debugging geometry data. The analyst generally resorts to sparse geometry only for very complex configurations where the dose versus thickness curve is steep.

2.4 EMBEDDING

The term embedding refers to the overlap of volume elements. Most complex geometry codes do not permit embedding and treat it as an error when detected. Two space radiation transport codes, MEVDP⁽¹⁹⁾ and HSV^(10, 12), do permit embedding and encourage it's use as a labor saving device. It is significant that these two codes permit sparse geometry while other codes require dense geometry.

MEVDP and HSV treat embedding in different ways; MEVDP uses local embedding whereas HSV uses global embedding except when it is treating MEVDP data. A brief description of the codes will illustrate the difference between local and global embedding.

In MEVDP the volume elements are simple convex geometric figures: ellipsoid truncated by 0, 1, or 2 planes; circular cylinder truncated by 2 planes; sphere; hemisphere; circular cone (1 nappe) truncated by 1 or 2 planes; and hexahedron as shown in Figure 3. Truncating planes are perpendicular to a symmetry axis. These volume elements are called elemental volumes or shields in the user's manual. Each shape may be void or "solid". Gases and liquids are termed solid here. Groups of 1 to 10 volume elements are associated to form an entity called a composite shield. Solid volume elements are specified first within each composite shield. Each solid may contain a different material but overlapping is generally avoided. Overlapping of solids is not embedding because penetrations through the common region are summed. Void volume elements are specified last within a composite shield. Voids embed the solids within the same composite shield (local embedding). However, such voids do not embed solids within other composite shields. Note that each volume element is a simple, convex figure, but a composite shield may be quite complicated in shape and composition.

In HSV volume elements are bounded by 1 to 50 surfaces of the following types: plane, ellipsoid, elliptic cylinder, elliptic cone, and toroid. Each volume element is homogeneous material or void. Thus, HSV volume element shapes may be quite complex including disjoint and multiply connected segments provided the polarity criterion is satisfied. Embedding is global in normal HSV usage. The term global means that if two or more volume elements overlap, the one occurring earlier in the data takes precedence in specifying material for the overlap region. However, HSV possesses a special mode to

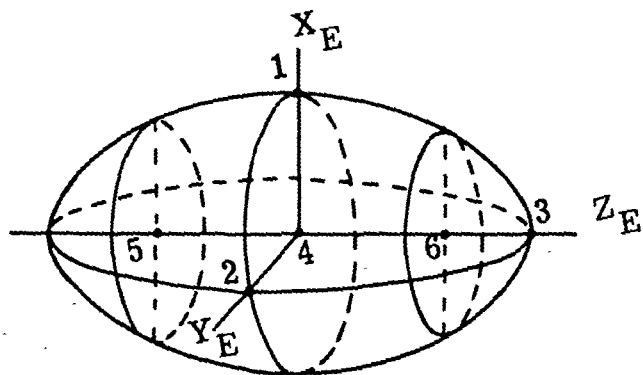
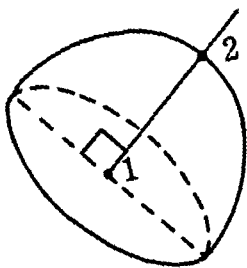
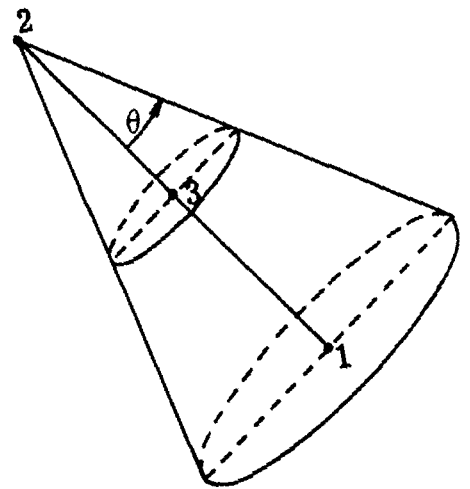
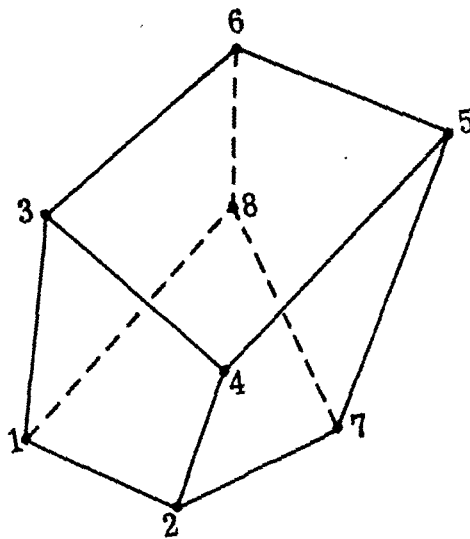
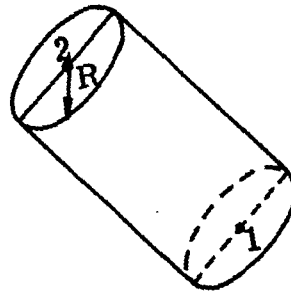
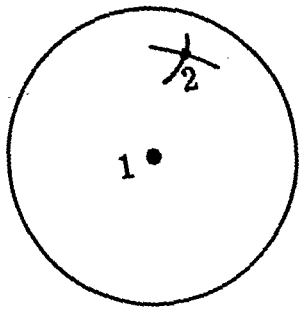


Figure 3. MEDVP Volume Elements.

process MEVDP geometry models intermingled with its regular data. In this case, the local embedding permitted in MEVDP is used within blocks of MEVDP data. Global embedding is still enforced among all solid volume elements.

A simple example will illustrate the use of the two codes. Consider a cylindrical aluminum tank with insulation on the outside and liquid on the inside as shown in Figure 4. Volume elements are simply labeled 1, 2, and 3. Number 1 encases the liquid; number 2, the tank; and number 3, the insulation. For MEVDP three composite shields are specified.

- Composite shield 1
 - Volume element 1, material 1

- Composite shield 2
 - Volume element 2, material 2
 - Volume element 1, void

- Composite shield 3
 - Volume element 3, material 3
 - Volume element 2, void

The ordering of composite shields is arbitrary, but the ordering of volume elements within these composite shields is fixed.

For HSV three volume elements are specified.

- Volume element 1, material 1
- Volume element 2, material 2
- Volume element 3, material 3

Here the ordering of volume elements is fixed because volume element 1 is (globally) embedded in 2 and 3, while 2 is embedded in 3.

This example illustrates the difference between local and global embedding. Geometry codes without the embedding feature such

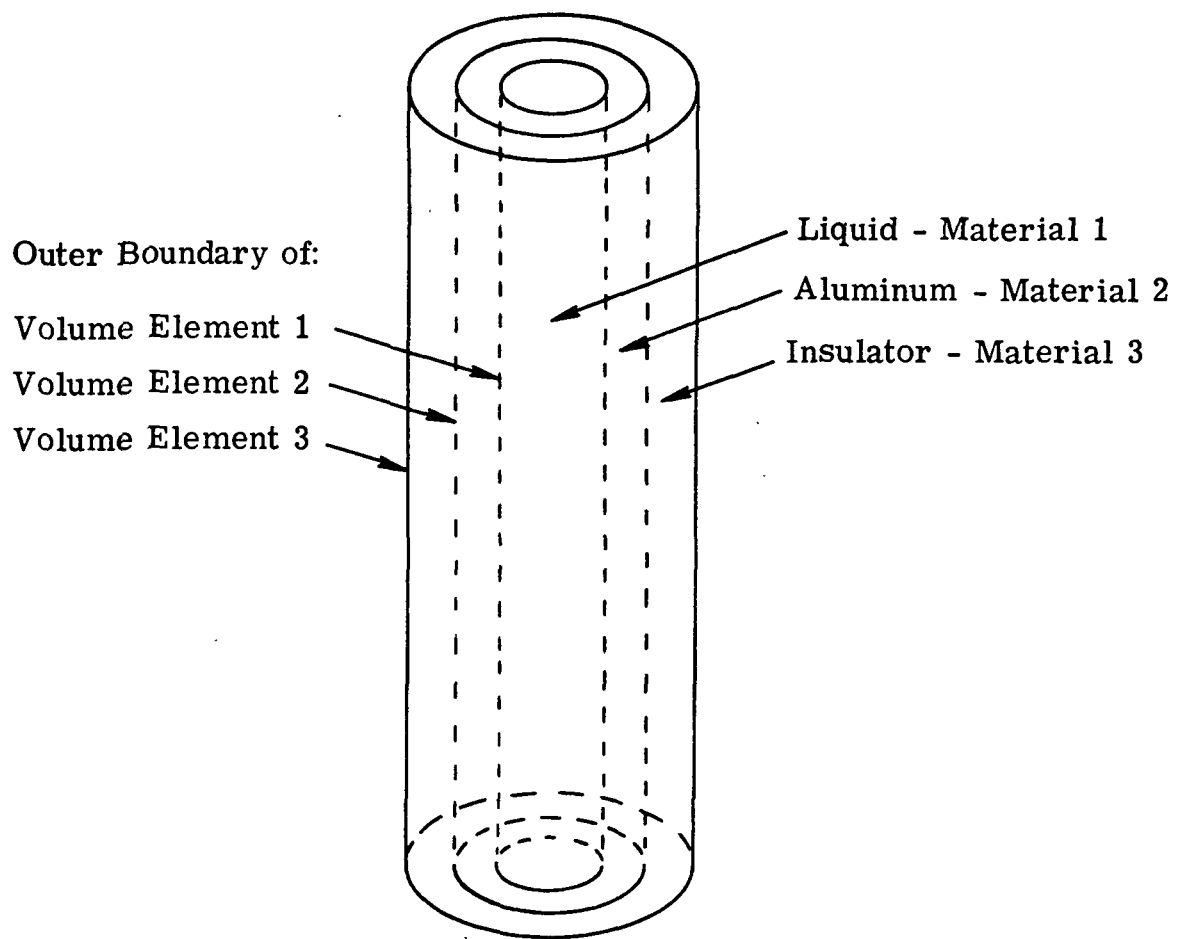


Figure 4. Insulated Tank.

as SIGMA can also model this simple example but would require seven volume elements.

2.5 TRANSFORMATIONS

Transformations are powerful aids in modeling complex configurations. A transformation consists of a rotation followed by a translation. HSV permits two levels of transformation. To illustrate their use, consider the following example.

A configuration to be modeled consists of a film vault containing 50 identical film cassettes. The vault is located within a space vehicle. This arrangement may be modeled in several ways. First, the difficult way is to compute all the volume element parameters in the vehicle coordinate system. Second, an easier way is to model the vault and one cassette in the vehicle system. Forty nine copies of the cassette are then made and moved to their proper positions with simple translations. Third, the easiest way is to model the vault in a convenient coordinate system (coordinate planes parallel to vault faces) and one cassette in either the vault system or its own coordinate system. Forty nine cassette copies are made and all 50 are moved to their proper place within the vault in the vault system by first level transformation. The vault and its contents are moved to the proper place in the vehicle system by a second level transformation. Should the location of the vault be changed (which happened numerous times in the Skylab program) the vault and its contents may be moved by changing one card containing the second level transformation in the HSV data.

MEVDP does not permit transformations for general composite shields. However, MEVDP will treat up to ten copies of an

accompanying man model by means of a double transformation. The first transformation may be used to convert the man model from the standing position to a sitting or prone position. Arms and legs may be rotated. The second transformation rotates and translates the model into the desired location in the vehicle.

It should be noted that transformations may be applied to MEVDP geometry data by writing an auxiliary program to process selected portions of the data.

Transformations of the MEVDP and HSV geometry data are relatively simple because the user is required to specify only a few points and parameters for each surface. The program may transform the points to a new coordinate system before it computes the algebraic coefficients of each surface.

The transformations themselves may be derived in a standard way. Let (X_1, Y_1, Z_1) denote a convenient coordinate system for data preparation. These data are to be transformed to a second convenient coordinate system, (X_2, Y_2, Z_2) , and finally to the vehicle system (X_v, Y_v, Z_v) . Either or both of the transformations may be omitted.

Let (A_1, B_1, C_1) be the coordinates of a point in the (X_1, Y_1, Z_1) system which is to be moved to (A_2, B_2, C_2) in the (X_2, Y_2, Z_2) system. The transformation is

$$\begin{pmatrix} A_2 \\ B_2 \\ C_2 \end{pmatrix} = R_{12} \begin{pmatrix} A_1 \\ B_1 \\ C_1 \end{pmatrix} + T_{12} \quad (5)$$

The second transformation moves the point to (A_v, B_v, C_v) in the vehicle system

$$\begin{pmatrix} A_v \\ B_v \\ C_v \end{pmatrix} = R_{2v} \begin{pmatrix} A_2 \\ B_2 \\ C_2 \end{pmatrix} + T_{2v}$$

R_{12} and R_{2v} are rotation matrices and T_{12} and T_{2v} are translation vectors. The components of T_{12} are simply the (X, Y, Z) coordinates of the origin of the first system in the second system (X_{10}, Y_{10}, Z_{10}) . Similarly, the components of T_{2v} are the (X, Y, Z) coordinates of the origin of the second system in the vehicle system, (X_{20}, Y_{20}, Z_{20}) .

The rotation matrices are constructed in the following manner. Let $\alpha_{11}, \beta_{11}, \gamma_{11}$ be the direction cosines of the X_1 axis in the (X_2, Y_2, Z_2) system. Similarly, $\alpha_{21}, \beta_{21}, \gamma_{21}$ and $\alpha_{31}, \beta_{31}, \gamma_{31}$ are the direction cosines of the Y_1 and Z_1 axes, respectively, in the second convenient system. Then

$$R_{12} = \begin{pmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} \\ \beta_{11} & \beta_{21} & \beta_{31} \\ \gamma_{11} & \gamma_{21} & \gamma_{31} \end{pmatrix} \quad (7)$$

Similarly, the rotation from the second convenient system to the vehicle system may be written

$$R_{2v} = \begin{pmatrix} \alpha_{12} & \alpha_{22} & \alpha_{32} \\ \beta_{12} & \beta_{22} & \beta_{32} \\ \gamma_{12} & \gamma_{22} & \gamma_{32} \end{pmatrix}$$

where $\alpha_{12}, \beta_{12}, \gamma_{12}$ are the direction cosines of the X_2 axis in the vehicle system, etc.

These transformations are usually easy to set up. For example, assume the axes of the first convenient system are parallel to those of the second convenient system (no rotation) but the component is to be displaced plus 10 units in the X direction and minus 5 units in the Z direction.

In this case

$$R_{12} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$T_{12} = \begin{pmatrix} 10 \\ 0 \\ -5 \end{pmatrix}$$

Next assume the second convenient system is to be rotated by an angle θ about the Z axis (positive rotation is counter-clockwise from above) and moved plus 50 units in X, minus 20 units in Y, and plus 20 units in Z to place it in the vehicle system. In this case

$$R_{2v} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$T_{2v} = \begin{pmatrix} 50 \\ -20 \\ 20 \end{pmatrix}$$

Occasionally the rotations are more tedious to compute than the simple examples given above. It is worth remembering that such calculations are still enormously easier than modeling tilted components directly in the vehicle system.

Transformations may be used for another labor-saving purpose. Assume two components to be modeled are identical except that one is a mirror reflection of the other, say in the Y direction. One may be modeled and the data deck duplicated. The duplicate deck may then be reflected by an improper rotation, R:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and moved to the correct position with a displacement vector T.

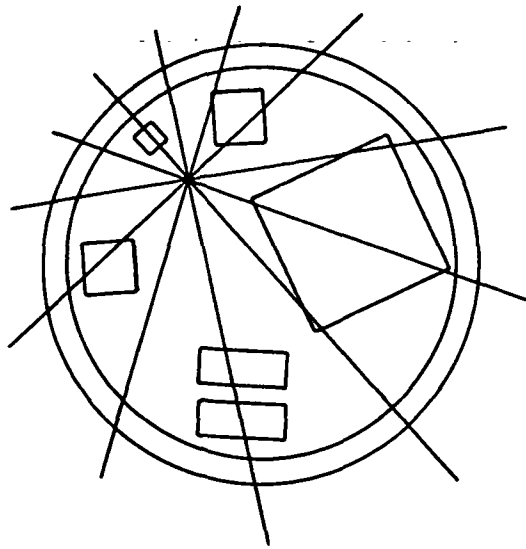
3.0 GEOMETRY MODEL SCANNING

This section describes the manner in which ray tracing may be performed in order to discover the shielding afforded a point within the configuration. The three space radiation transport codes in widest use - MEVDP, HSV, and SIGMA - are point kernel codes which assume straight line paths. Justification for the proton straight-ahead approximation is contained in another report⁽²⁾. Figure 5 shows a two dimensional representation of the three dimensional ray tracing in a typical configuration. The dose at point A is desired. The geometry program sets up an array of vectors emanating from Point A (often called a detector, receiver, or dose point), in a manner discussed later, in order to scan the configuration. The material encountered by each ray is assumed to be typical of the solid angle about that ray. The transport calculation is then performed in a spherical shell shield geometry, Figure 5 b, equivalent to the original configuration of Figure 5 a. The problem area discussed in this section concerns the generation of rays and the efficient tracing of rays through the configuration.

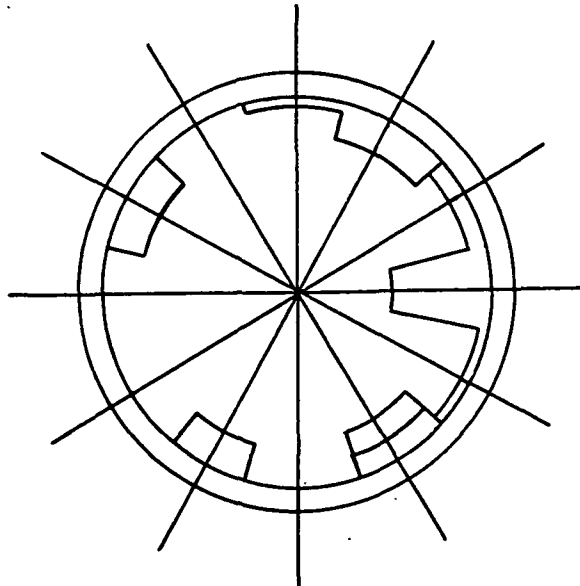
It should be noted that at least one space radiation transport code does not use point kernel methods. Program BETA⁽¹⁵⁾ performs Monte Carlo estimates of electron transport through complex configurations using the SIGMA complex (dense) geometry subroutine. The techniques discussed below generally will not apply to such codes.

3.1 RAY GENERATION

The method chosen to generate ray directions is controversial. The choice lies between stochastic sampling or some sort of systematic sampling. Langley and Billings state that systematic sampling may



a. A Ray Tracing



b. Equivalent Shield

Figure 5. Representation of Ray Tracing.

lead to serious bias. For example, the method chosen might consistently hit or miss symmetrical bulkheads. Their SIGMA code can sample ray directions stochastically or systematically, but they recommend stochastic sampling. On the other hand, Liley states that: 1) Stochastic sampling sometimes leads to increased variance which can only be reduced by increasing the number of rays, and 2) Situations where systematic sampling can cause serious errors are usually obvious. Minor adjustments in the systematic sampling technique can avert serious bias. Liley's code, MEVDP, can use either technique but he usually uses systematic sampling of ray directions. The HSV code uses systematic sampling of ray directions. In special situations, the HSV geometry model is rotated to minimize bias.

3.1.1 Stochastic Ray Direction Sampling

Ray directions may be sampled from a distribution giving equal probabilities in all directions. Let θ and ϕ represent polar and azimuthal angles in the spherical coordinate system of Figure 6. R1 and R2 are random numbers with a uniform distribution in the range ($0 < R < 1$) from a good pseudorandom number generator. The angles may then be computed as:

$$\theta = \cos^{-1} (2 \cdot R1 - 1), \quad \phi = 2\pi \cdot R2$$

If N rays are selected in this manner, the solid angle, $\Delta\Omega$, associated with each ray is:

$$\Delta\Omega = \frac{4\pi}{N}$$

A foreknowledge of the configuration might make it desirable to sample some directions more carefully than others to ensure thorough scanning of thinly shielded directions. Because many schemes are feasible, the reader is referred to Kahn ⁽¹⁸⁾ for suggestions on biased sampling procedures and weight corrections. A sample procedure is outlined below.

Suppose a configuration is lightly shielded near the plus Z axis. Examination shows that a cone centered on the Z axis, with vertex at the detector point and a half angle of 60° , includes the thin section. Then the polar angles of N_1 rays may be chosen within the cone as follows:

$$\theta_1 = \cos^{-1} \left(\frac{R+1}{2} \right)$$

where R is a pseudorandom number. The polar angles of N_2 rays in the remaining solid angle may be chosen as follows:

$$\theta_2 = \cos^{-1} \left(\frac{3R-2}{2} \right)$$

Azimuthal angles are sampled from Equation 9. The solid angles represented by each ray are:

$$\frac{\pi}{N_1} \text{ steradians inside the cone}$$

$$\frac{3\pi}{N_2} \text{ steradians outside the cone.}$$

The user may vary the scanning in each region by a suitable choice of N_1 and N_2 .

Should direction cosines (l, m, n) in a Cartesian system be preferred in place of the above angles, the conversion is

$$\begin{aligned}l &= \sin \theta \cos \Phi \\m &= \sin \theta \sin \Phi \\n &= \cos \theta\end{aligned}$$

3.1.2 Systematic Ray Direction Sampling

Ray directions may be sampled systematically in many ways. The methods used in MEVDP and HSV will be outlined as illustrations.

The MEVDP system uses equal increments in $\cos \theta$ and equal increments in Φ to obtain equal solid angles. Angles θ and Φ are spherical coordinates shown in Figure 6.

Let N_{θ} = number of $\cos \theta$ increments

N_{Φ} = number of Φ increments.

Then $\Delta \cos \theta = \frac{1}{N_{\theta}}$

and $\Delta \Phi = \frac{2\pi}{N_{\Phi}}$

are the computed increments. Rays are taken from the origin centered at the dose point to the center of each solid angle increment.

Ray generation in HSV is somewhat more complicated. An axially symmetric figure is generated by rotating line segments about the Z axis of a coordinate system centered at the dose point. Each rotated line segment generates a truncated conical or cylindrical shell. Each shell is approximated by six equal planar

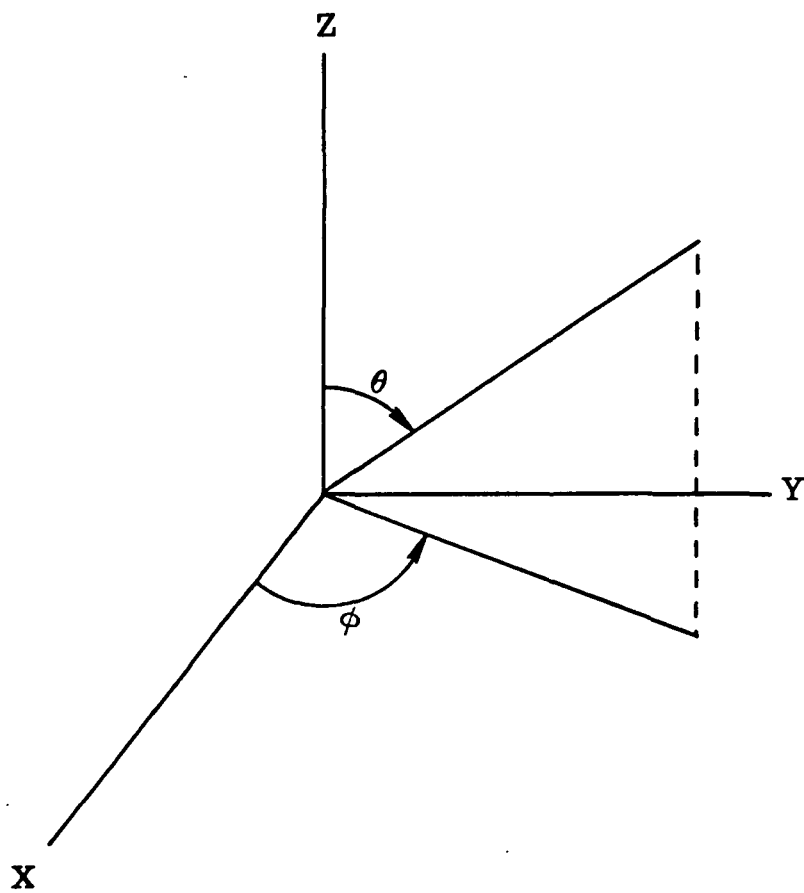


Figure 6. Ray Spherical Coordinates

facets. A portion of this subdivision is shown in Figure 7.

Each facet is triangular or trapezoidal. The solid angle subtended by each facet from the dose point is computed and compared to a user-specified solid angle criterion (SAC) for that facet. If the solid angle is larger than the SAC for that facet, the facet is repeatedly subdivided until the criterion is met. At that stage, a ray is constructed from the dose point to the centroid of the (subdivided) facet.

Note that rays do not necessarily represent equal solid angles. The user may choose to scan certain directions more closely by decreasing the SAC for appropriate facets, thus forcing finer subdivision. Unlike the MEVDP method, the number of rays calculated by HSV is not known in advance. For the case where all SAC's are equal, a rule of thumb is:

$$\text{Number of rays} \approx 1.8 \left(\frac{4\pi}{\text{SAC}} \right)$$

For example, if all SAC's are 0.2 steradians, the number of rays generated is usually 108, which is close to the value of 113 obtained from the above formula.

3.2 RAY TRACING

Ray tracing is the objective of a complex geometry program. It is tedious and expensive. Nearly all such programs incorporate schemes to speed the process and reduce computer time. In this section, the calculation of volume element penetrations is illustrated. Volume elements are bounded by surfaces, so the first step is to compute the intersection of a ray and a surface.

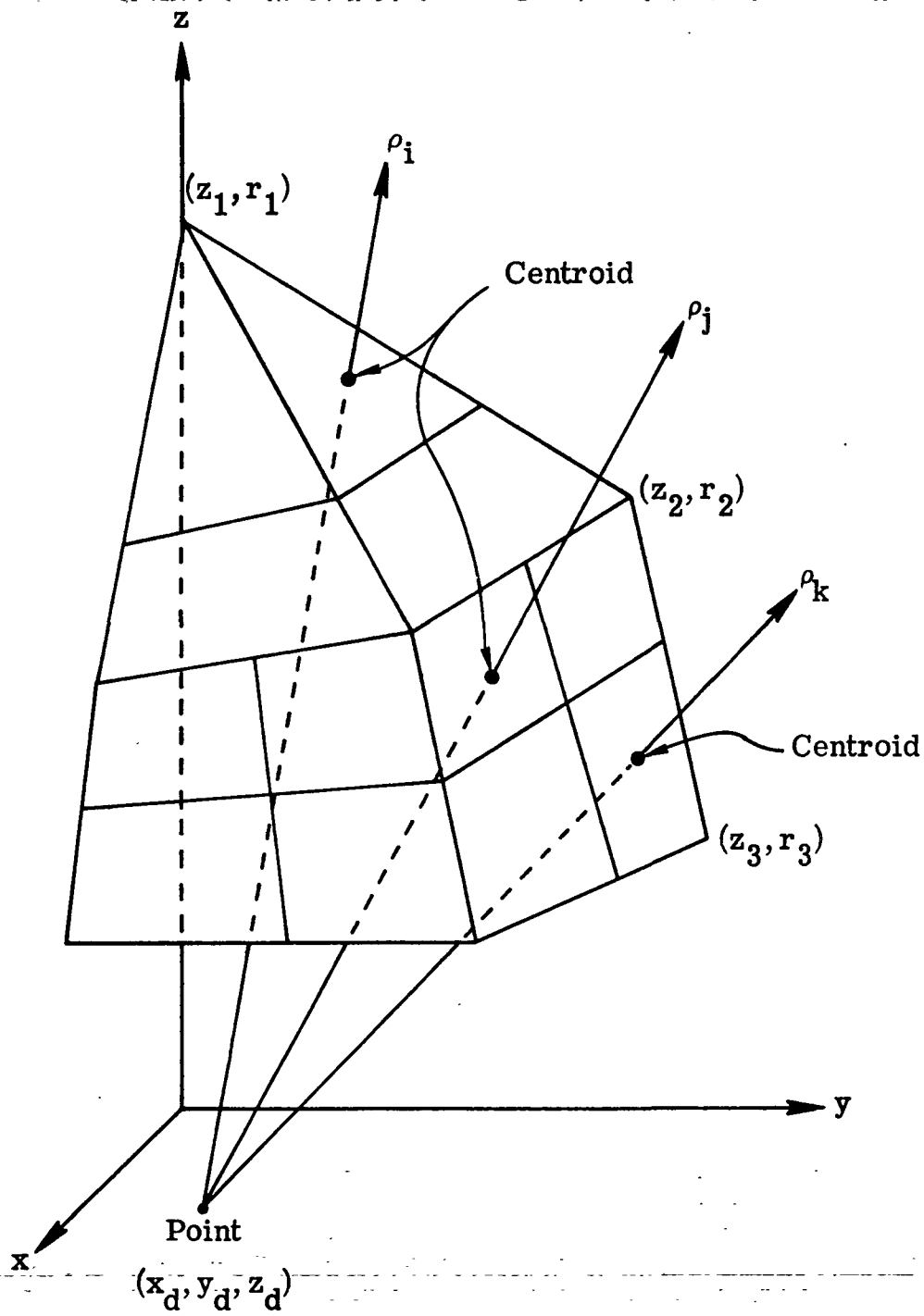


Figure 7. Polyhedral Envelope for HSV¹⁰

Consider a ray originating at a dose point (X_d, Y_d, Z_d) and proceeding in a direction given by direction cosines (l, m, n) . Let the distance to one of the surfaces bounding a volume element be ρ , as yet undetermined. Label the intersection point, as yet undetermined (X_i, Y_i, Z_i) . The coordinates of the intersection may be written as:

$$X_i = X_d + l \rho$$

$$Y_i = Y_d + m \rho$$

$$Z_i = Z_d + n \rho$$

Intersections with several types of surfaces will be considered.

Plane Surface - The intersection point (X_i, Y_i, Z_i) lies on the plane and the following equation is satisfied.

$$AX_i + BY_i + CZ_i - D = 0$$

Substituting Equations 12 into 13,

$$A(X_d + l \rho) + B(Y_d + m \rho) + C(Z_d + n \rho) - D = 0$$

so the distance, ρ , to intersection is

$$\rho = \frac{D - AX_d - BY_d - CZ_d}{Al + Bm + Cn}$$

A zero denominator or a negative ρ implies no intersection. The intersection point coordinates are calculated from Equations 12.

Quadric Surface - The intersection point (X_i, Y_i, Z_i) on the quadric surface and the following equation is satisfied.

$$\begin{aligned}
& AX_i^2 + BY_i^2 + CZ_i^2 + DX_iY_i + EX_iZ_i + FY_iZ_i \\
& + GX_i + HY_i + PZ_i + Q = 0
\end{aligned}$$

Again, Equations 12 are substituted into Equation 15. The resulting equation is solved for ρ :

$$\rho = \frac{-U \pm (U^2 - 4VW)^{1/2}}{2V}$$

where

$$V = Al^2 + Bm^2 + Cn^2 + Dlm + Eln + Fmn$$

$$\begin{aligned}
U = & 2(AX_d l + BY_d n + CZ_d n) \\
& + D(Y_d l + X_d m) + E(Z_d l + X_d n) \\
& + F(Z_d m + Y_d n) + Gl + Hm + Pn
\end{aligned}$$

$$\begin{aligned}
W = & AX_d^2 + BY_d^2 + CZ_d^2 + DX_d Y_d + EX_d Z_d \\
& + FY_d Z_d + GX_d + HY_d + PZ_d + Q.
\end{aligned}$$

If V is zero, as may be happen certain circumstances with the cone, then

$$\rho = \frac{-W}{U}$$

Complex solutions imply a miss by the ray. The intersection point coordinates are then calculated from Equations 12.

Toroidal Surface - The intersection of a ray and a toroidal surface is tedious to compute. The method will be outlined here with most of the algebra omitted for brevity.

The equation of a torus in a canonical coordinate system with the Z axis as the axis of rotation is:

$$AZ^2 + B[(X^2 + Y^2)^{1/2} - C]^2 = D$$

Transformation of this equation to the vehicle coordinate system would produce a 35 term equation. Storage space and computer time are saved by transforming the dose point and ray direction cosines into the canonical system. The right sides of Equations 12 are transformed to the canonical system and substituted into Equation 18. The value of ρ is obtained and put into (untransformed) Equations 12 to obtain the intersection points. This process invokes the solution of a quartic equation which requires double precision. The number of intersections may be 0 to 4.

For sparse geometry codes, the calculation of path lengths through volume elements proceeds as follows. The intersection points, p_i , with all surfaces bounding a volume element are computed and arranged in order of increasing distance. Negative distances are removed. Some intersections may be false; e.g., the ray may strike several planes bounding a hexahedron but miss the hexahedron itself. The polarity criterion is used to screen for real intersections. The dose point is added at the beginning of the ordered list of intersections with a distance of zero. Starting with $i = 1$, the coordinates of a point (X_m, Y_m, Z_m) midway between p_i and p_{i+1} are computed and substituted into the polarity equation for each surface in turn. To reiterate; these equations are:

$$AX_m + BY_m + CZ_m - D = q \quad (\text{Plane})$$

$$AX_m^2 + BY_m^2 + CZ_m^2 + DX_m Y_m + EX_m Z_m + FY_m Z_m \\ + GX_m + HY_m + PZ_m + Q = q \quad (\text{Quadric})$$

$$AZ_m^{*2} + B[(X_m^{*2} + Y_m^{*2})^{1/2} - C]^2 - D = q \quad (\text{Toroid})$$

Here, the asterisk denotes transformation into a canonical system.

If the q 's are positive for all surfaces bounding a volume element, the path length between p_i and p_{i+1} is a real penetration and is saved. The dose point is included in the list in case it is inside the volume element.

A code requiring convex volume elements would stop after one penetration was discovered. Codes permitting reentrant surfaces check all pairs of intersection points, saving those for which all q 's of the polarity criterion are positive. Each volume element is subjected to the above search.

Codes which permit embedding perform an additional check at this point. MEVDP pauses at the conclusion of each composite shield to subtract out those penetration lengths where voids overlap solids (local embedding). HSV keeps a table of the penetration start and end points, corresponding volume element number, etc. After all volume elements are processed for a particular ray, a check is made for overlaps (global embedding). Where overlap is discovered, the lowest volume element number is assigned to the segment.

Thus a sparse geometry code must check every volume element for intersections with each ray. This process is very lengthy and provides incentives for quick screening tests which will be described in section 3.4.

3.3 EFFICIENCY SCHEMES FOR DENSE GEOMETRY RAY TRACING

Dense geometry codes permit much faster searches than non-dense codes. Usually a dense geometry code requires the user to input surface coefficient data for up to 100 surfaces depending on the configuration. A separate table of volume element characteristics is prepared. These characteristics include a list of surfaces bounding each volume element and parameters establishing the polarity criterion for that volume element. The code may then prepare an inverse list automatically, specifying all volume elements which touch each surface. If this is done, ray tracing is quite fast. At each boundary crossing, a restricted list of potential next volume elements is available to the code and many need not be tested at all.

A dense geometry program can be designed to even greater efficiency. Usually a ray leaving a particular volume element has access to only a few adjacent volume elements. The code can keep track of next volume elements from previous rays exiting each volume element and test that priority list first.

Because sparse geometry codes must check every volume element (unless tricks are used) while dense geometry codes can quickly find the paths through a configuration, a great disparity may exist in computing speed. Given the disparity in speed between dense and sparse geometry codes, why should anyone choose a sparse

geometry code to solve a problem? The primary reason is necessity. Current dense geometry codes cannot treat complex configurations such as the Gemini vehicle, the Apollo CSM, the Lunar Module, and the Skylab cluster. Increased computer capabilities now permit treatment of sparse geometry at reasonable cost. Even treating a configuration of several thousand volume elements, a UNIVAC 1108 can trace one to ten rays per second. Thus a 200 ray trace would require only a few minutes of computer time.

3.4 EFFICIENCY SCHEMES FOR SPARSE GEOMETRY RAY TRACING

The techniques illustrated in this section are principally derived from three sources. Oden (of Sweden) has reported a code⁽²²⁾ which treats fragmentation and ricochets from armored vehicles. MEVDP and HSV use several of the same scanning methods as a result of independent developments. Perhaps the referenced documents, together with this report, will serve as a starting point for the next generation of sparse geometry codes.

3.4.1 Oden's Scanning Techniques

The volume elements used in Oden's code are slightly different from those described previously. The differences are minor and will not be discussed here. The scanning is subdivided into three classes; quick scan, fine scan, and detail scan. Within each class, it is not evident whether all tests are applied.

Under the quick scan tests, Oden lists the sphere test, the vertical plane including the ray test, and the polyhedron with axis parallel to the ray test. The sphere test is not elucidated but is probably similar to that used in MEVDP and HSV. Essentially a sphere circumscribing the volume element is constructed.

If the ray misses the sphere, it will miss the volume element. If the ray hits the sphere, a detailed check must be made to see whether it hits the volume element.

The vertical plane including the ray test is not described in detail in Oden's report. It may involve rotation of the coordinate planes till a vertical plane includes the ray. In the rotated system a miss would be signaled by all like coordinates of a volume element having the same sign. The polyhedron with an axis parallel to the ray test is similar to a test in MEVDP; it will be discussed in the next section.

Under fine scan tests, Oden lists:

- 1) XYZ limits for volume elements, ray start and end points,
- 2) Angular limits for volume elements,
- 3) Block listing of volume elements, and
- 4) Ordered embedding.

Under item 1), the extreme values X, Y, and Z of the volume element are listed. The starting point of a ray is given, the end point refers to the place where it enters an external void. A simple check reveals whether the ranges of each coordinate overlap for the ray and the volume element. If any coordinate fails to overlap, the volume element is missed. If all three overlap, a more detailed check is required. Item 2, angular limits, is a test used for hexahedrons, cylinders, and cones in MEVDP. It will be discussed in the next section. Item 3, block listing of volume elements, refers to subdivision of the configuration blocks. Volume elements wholly contained within each block are so listed. If a ray misses the block, it will miss those volume elements. This technique is reminiscent of the octant test in MEVDP and the

super region test in HSV to be discussed in the next sections.

Item 4, ordered embedding, refers to complete embedding of some volume elements within others. The larger ones are tested first. If they are missed by a ray, contained volume elements will be missed also.

The detailed scan is relatively straightforward. The positive distance to entering and leaving points is computed with the aid of the polarity criterion. Then the maximum distance entering and minimum distance exiting points are selected. This max-min segment is tested against all surfaces with the polarity criterion to determine whether it is inside the volume element. This test applies to convex volume elements.

3.4.2 MEVDP Scanning Techniques

MEVDP uses several fast scan techniques to screen out some volume elements from detailed checking for ray intersections. The fast scan methods are applied to each volume element in turn. Void volume elements within a composite shield are skipped if no solid volume elements have been hit.

The first scan test (applied to all types of volume elements) is the octant test. Survivors are then subjected to either a sphere test, or a projection test. Survivors of the second test are subjected to a detailed ray intersection check.

It is necessary to understand three of the coordinate systems used in MEVDP before assessing the fast scan tests. The Absolute Coordinate System (ABCS) is the only one devised

by the user. The ABCS is a righthanded Cartesian system in which the user defines volume elements and dose points. The Dosimeter Coordinate System (DSCS) is automatically computed by translating the origin of the ABCS to the current dose point as shown in Figure 8. The translation for an arbitrary point is:

$$X_D = X_A - X_{AD}$$

$$Y_D = Y_A - Y_{AD}$$

$$Z_D = Z_A - Z_{AD}$$

where

(X_A, Y_A, Z_A) are the coordinates in the ABCS,

(X_D, Y_D, Z_D) are the coordinates in the DSCS, and

(X_{AD}, Y_{AD}, Z_{AD}) are the coordinates of the dose point.

After this translation the ray, of course, emanates from the origin and usually traverses one octant. No octant test is made if the ray lies within a coordinate plane. Otherwise a simple test is made to determine whether the volume element lies entirely within the same octant. If the ray and volume element are in different octants, a miss is registered. The octant test is bypassed if the volume element lies (or might lie) in two or more octants.

Those volume elements which pass or bypass the octant test are subjected to an additional test prior to a detailed intersection check. For the sphere, no transformation is necessary. A simple sphere intersection test is made in the DSCS. If the spherical volume element passes the sphere test, exact distances to intersection are calculated. The same test is applied to the hemisphere,

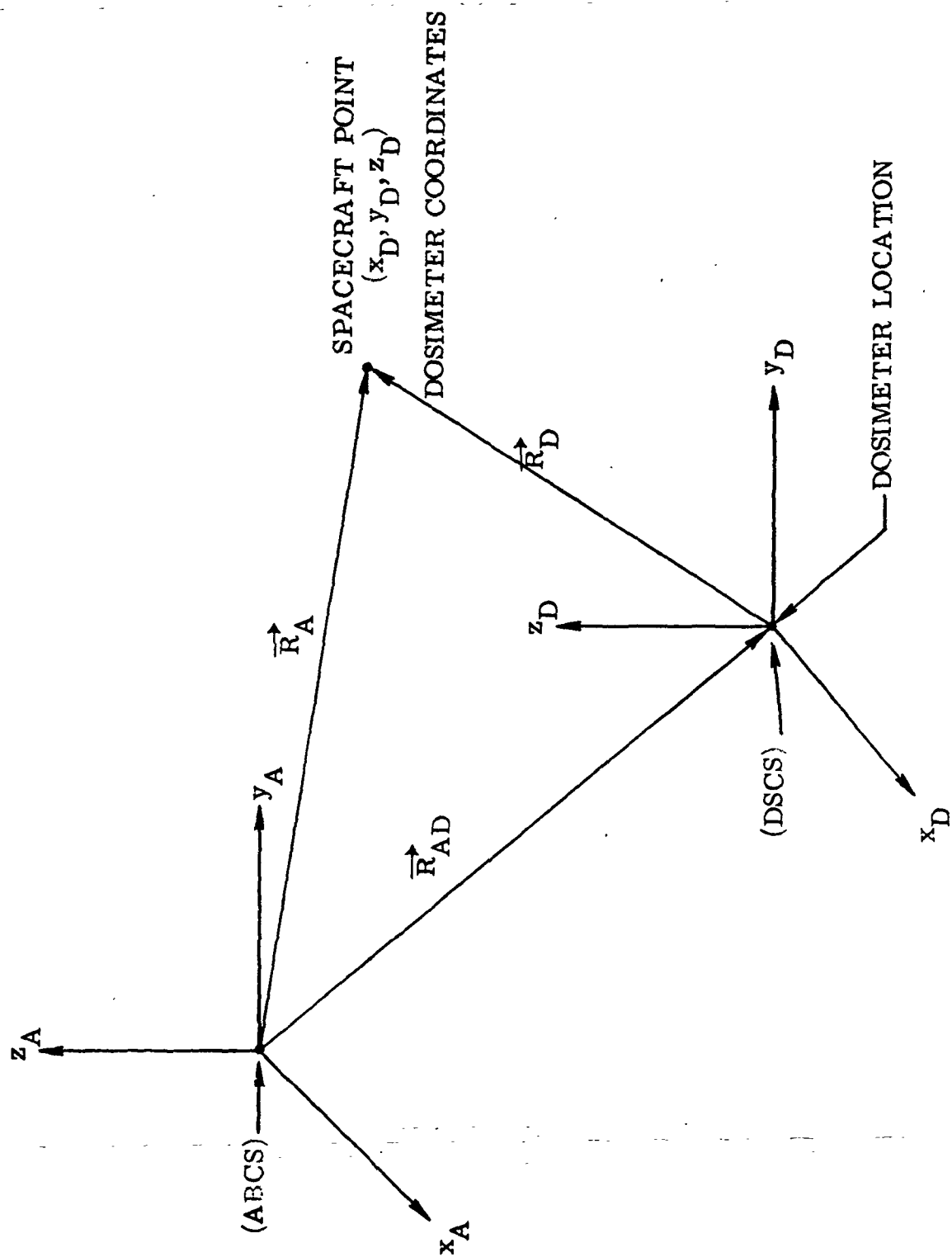


Figure 8. Dosimeter (DSCS) and Absolute (ABCS) Coordinate Systems for MEVDP¹⁹

except that a hit of the sphere does not necessarily mean that the ray hits the hemisphere.

Other types of volume elements require a special coordinate system, the Rotated Detector Coordinate System (RDCS). For the hexahedron, a rotation is applied to place the Z-axis of the RDCS along the tracking ray. For the hemisphere, (truncated) cone, (truncated) ellipsoid, and the cylinder, a rotation is applied to place the Z-axis parallel to a symmetry axis of the volume element. The rotation angles for the cylinder are shown in Figure 9. Note that the first rotation, Φ about the Z(DSCS) axis moves the Y-axis until it is parallel with the projection of the cylinder axis in the XY plane. Then a rotation about the new X-axis makes the Z(RDCS) axis parallel with the cylinder axis.

It may be noted that one rotation matrix serves all rays for each hemispherical, conical, ellipsoidal, and cylindrical volume element. Also, one rotation matrix per ray serves all hexahedrons in the configuration.

For the ellipsoid, the code selects the largest principal semiaxis to construct an enveloping sphere for the sphere test.

The MEVDP sphere test for the cylinder and cone is erroneously explained on page 35 of reference 19, but is further elucidated on pages 83 - 84 of that document. Figure 10 shows the geometry, where it is understood that a (truncated) cone circumscribed by the cylinder shown may be considered. The projection of the cylinder on the XY plane (RDCS) is a circle. If the projection of the ray on the XY plane (RDCS) hits the circle,

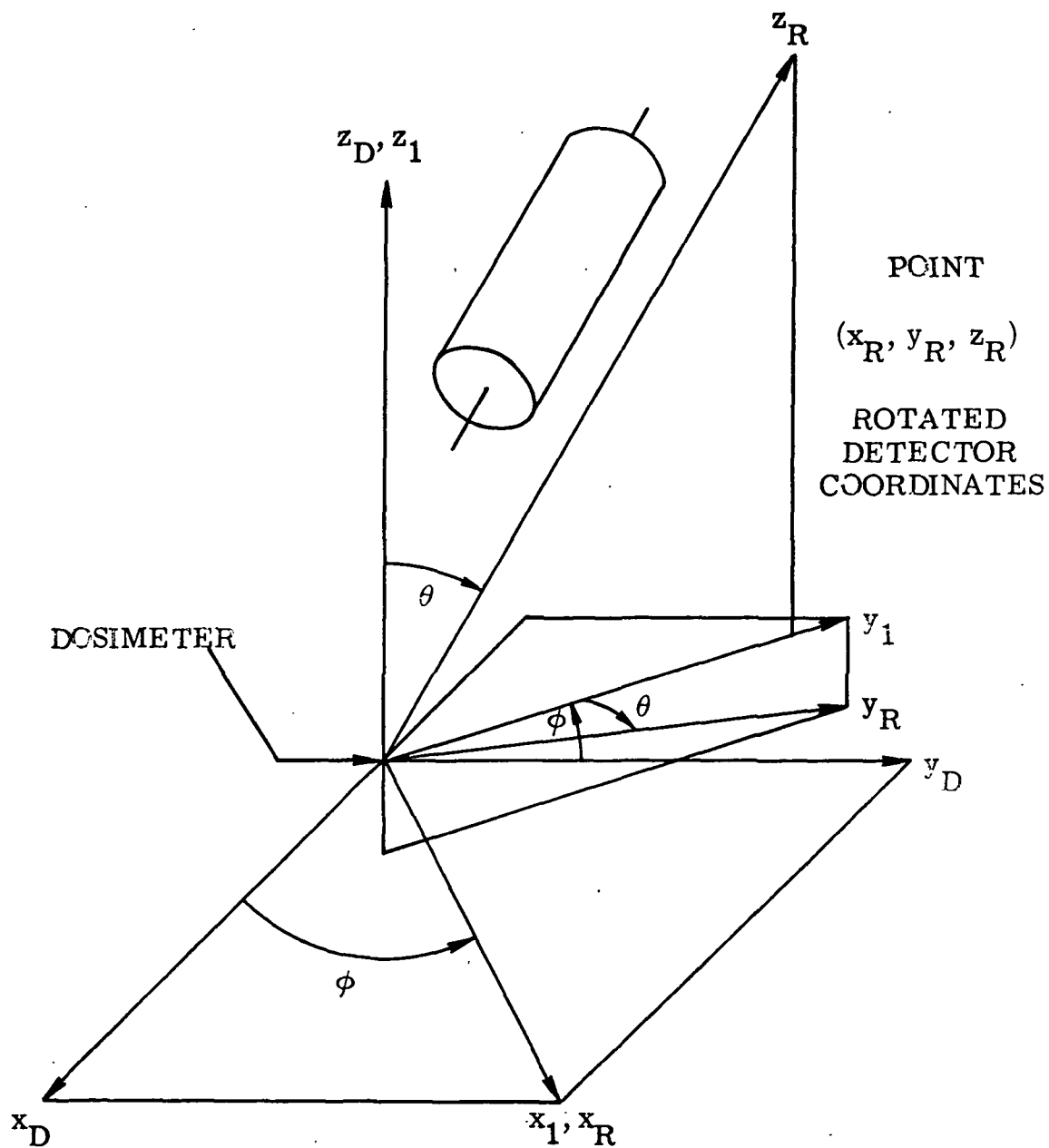


Figure 9. Rotated Detector Coordinate System (RDCS) for MEVDP¹¹

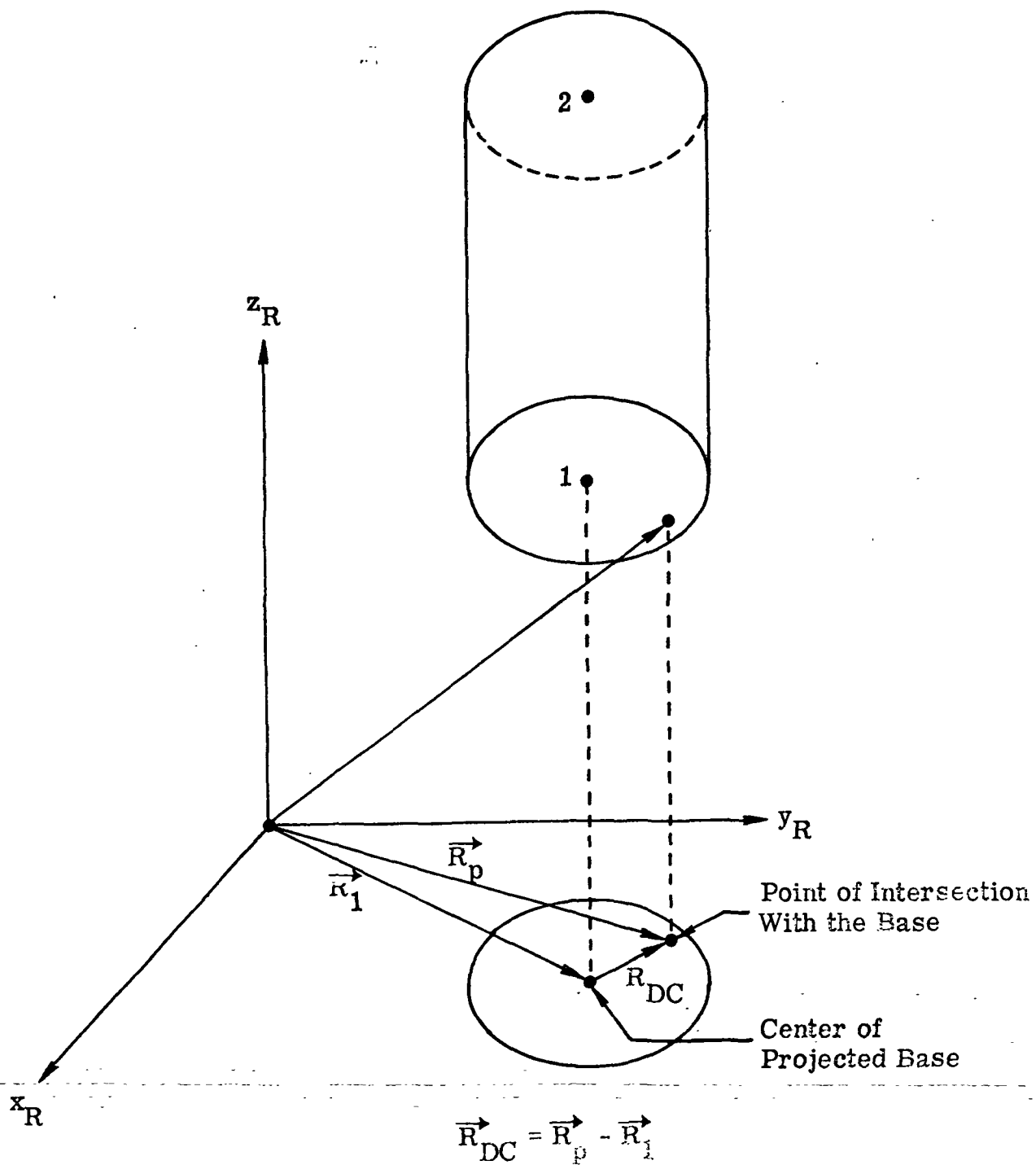


Figure 10. Projection of Cylinder (Cone) Base Plane for MEVDP¹⁹

the ray may or may not hit the volume element. If the projection of the ray misses the circle, the ray will miss the volume element. In this case, projected circle test or angle test appear to be better terms than sphere test.

Finally, a hit or miss test is applied to the hexahedron in the RDCS system. Because the Z-axis (RDCS) lies along the tracking ray for the hexahedron only, each face is projected into the XY plane (RDCS) as shown in Figure 11. If the projected figure includes the origin, the ray penetrates that face.

It is apparent that the translation to the DSCS system centered at the dose point leads naturally to the octant test. The rotation to the RDCS system allows the sphere test or the projected figure test, as well as detailed intersections, to be performed in a canonical system where the equations are in a simple form.

3.4.3 HSV Scanning Techniques

This code treats a more general geometry than codes discussed previously. A consequence of this fact is that incorporation of quick scan techniques is difficult without assistance from the user. The code has been described in 1964⁽²⁴⁾, 1965⁽⁸⁾, 1966⁽⁹⁾, and 1967⁽¹⁰⁾ without reference to quick scan techniques. Starting in 1967, the code was applied to a very complex configuration, Skylab, and the addition of quick scan methods became expedient.

3.4.3.1 HSV Super Regions

In 1967, the super region quick scan method was incorporated. A super region is a fictitious volume element bounded by an elliptic or circular cylinder and two truncating planes, not necessarily

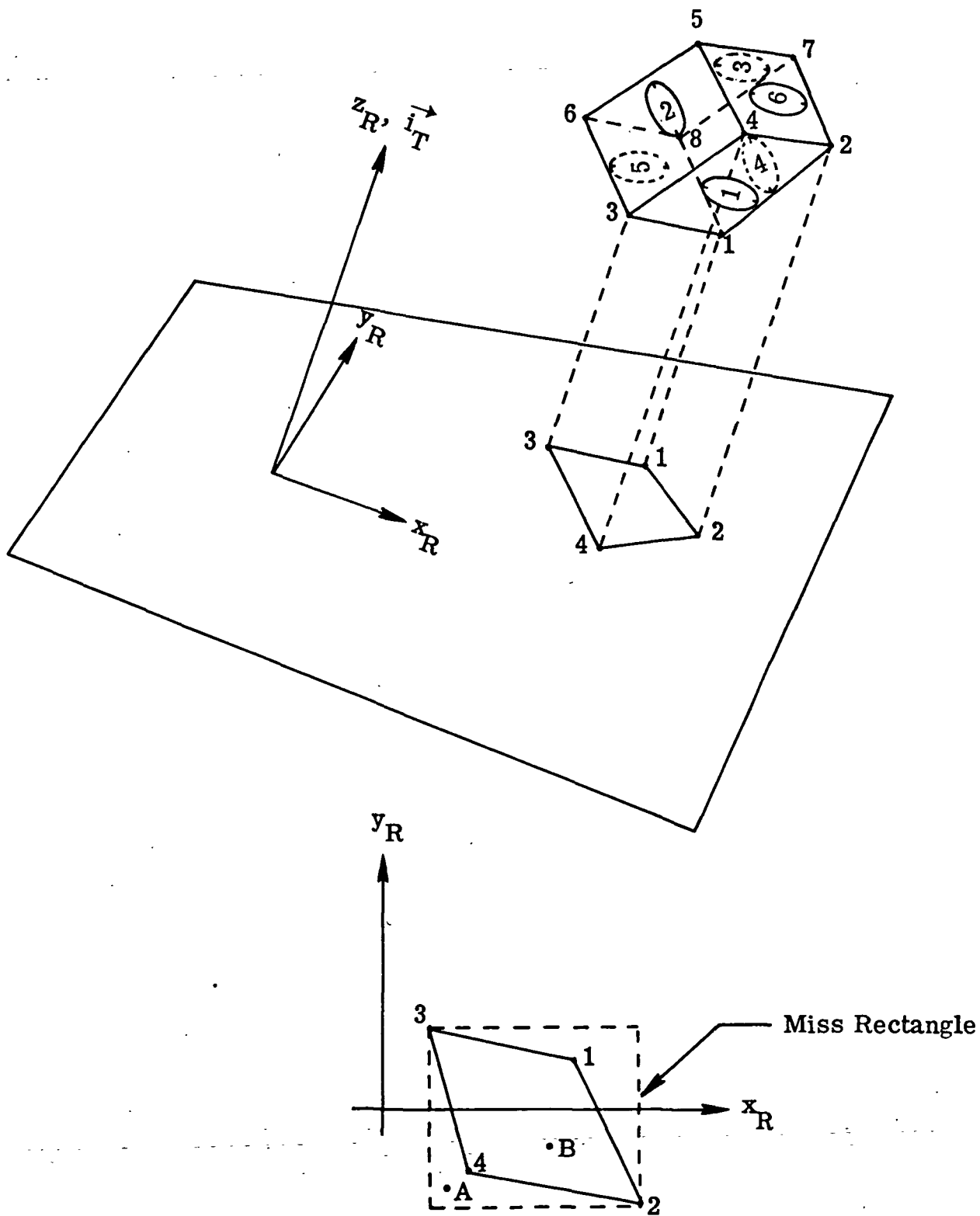


Figure 11. Projection of Hexahedron Face for MEVDP¹⁹

perpendicular to the cylinder axis. The super region is defined in the same manner as a normal volume element, except that a super region flag is set. All volume elements which follow must be included in that super region until it is turned off with another flag. In the ray tracing process, the code processes the super region first in that block of data. If the ray misses the super region, checks on the circumscribed volume elements are skipped.

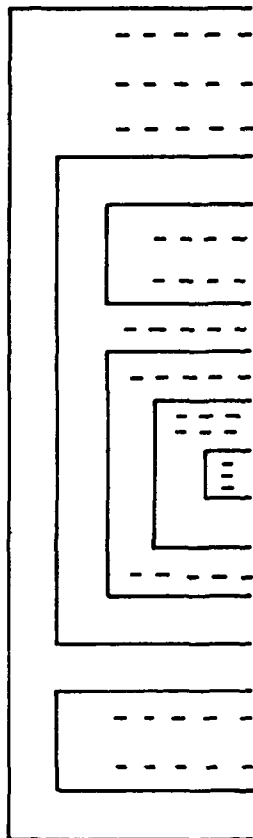
Nesting of super regions, similar to the nesting of DO loops in a FORTRAN program, is permissible to a level ten deep. The maximum depth used to date is six deep. A schematic of the nesting is shown in Figure 12.

An illustration of the nesting of super regions to a five level depth may be taken from the Skylab effort. A super region is placed around the Orbiting Work Shop (OWS). Within the OWS, a second super region is placed around the film vault. Within the vault, third level super regions are placed around each of the 12 drawers. Within a particular drawer, fourth level super regions are placed around each of four rows of film magazines. Within each row, fifth level super regions are placed around clusters of ten film magazines.

The super region technique is efficient if the user takes advantage of it. Early in the Skylab program, 80 super regions were added to a 2000 volume element configuration which was set up without super regions in mind. Computer run time was reduced by a factor of 20.

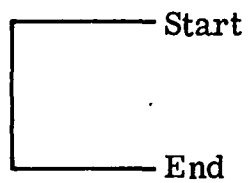
Super Region
Nesting Level

1 2 3 4 5



Code: ...

Super Region



Volume Element

Figure 12. Schematic of HSV Super Region Nesting.

3.4.3.2 HSV Sphere Test

In 1969, the Lunar Module and Apollo Block II Command and Service Module were added to the Skylab geometry model. Because these components had already been modeled in the MEVDP system, the HSV logic was changed slightly to permit use of MEVDP geometry models. Again running time increased to an undesirable extent and shortcuts were sought. Super regions could not be added easily because individual MEVDP volume elements were identified by number rather than name. However, the simple shape of each volume element did lend itself to a sphere test.

For MEVDP volume elements, HSV calculates a central point. The distance from that point to the furthestmost part of the MEVDP volume element is also easily computed. That distance serves as the radius of a circumscribing sphere centered at the central point. In the ray tracing process, intersection of the ray with the sphere is checked first. If the ray misses the sphere, it misses the volume element and a detailed check is unnecessary.

The sphere test worked so well with MEVDP data that it was added as an option to all other HSV geometric data. The user may specify a radius for a circumscribed sphere centered at the internal point for any volume element or super region.

User specification of the sphere radius leads to a possibility of error if the sphere is too small. The HSV code provides some error checking by taking 50 points just outside the surface of the sphere and using the polarity criterion on each point. If any one of these points is within the volume element, an error signal is generated.

3.4.3.3 HSV Torus Test

Determining ray intersections with a torus is such a lengthy calculation that a special quick scan test is included in the code. The user may, of course, assist the code with the super region and sphere test options for the volume element that uses the torus as one of its surfaces. However, if the code does need to check the toroidal surface (not necessarily the only surface of that volume element), a special torus test is made without user assistance.

The code derives surface coefficients for the circumscribing cylinder and tangent planes. If the ray misses the cylindrical figure, it misses the torus. If the ray hits both planes within the large cylinder, a second check is made to see if it passes through the small cylinder inscribed in the hole of the torus. If the ray intersection is not then rejected, a detailed check is made.

The super region test, the sphere test, and the torus test may be used to speed HSV ray tracing to the point where it is faster than MEVDP ray tracing. Both codes are slower than the dense geometry code, SIGMA.

4.0 DATA FORMAT

The term "data format" as used herein concerns the type of data required for a geometry program as well as the arrangement of data on input cards. The discussion below will touch upon data format requirements for SIGMA, MEVDP, and HSV. Detailed specifications may be found in the user manuals; simple descriptions will be used where feasible in this section.

The preparation of large geometric data sets for these codes is an arduous task. The user should keep a log referencing volume element, shape, material, density, blueprint number, and blueprint date or revision number. A concise word description in the card deck is also helpful.

4.1 SIGMA DATA FORMAT

The SIGMA code is restricted to dense geometries. A region of space is subdivided by surfaces into volume elements. Each point within the space is contained within one and only one volume element unless it is on a boundary. The user sketches the model, then numbers the surfaces and volume elements. The algebraic coefficients of each surface are computed by the user and placed on cards. Next, the volume elements are specified by listing the surfaces associated with each one. An internal point is given for each volume element to assist the code in setting up the surface polarity table.

The SIGMA code substitutes the internal point into each surface bounding a volume element. If the polarity is correct, the surface number is placed in the surface list for that volume element. If the polarity is wrong, a negative sign is placed with the surface number in the list for that volume element. An example will illustrate. Consider

the cube centered at (1, 1, 1) with sides equal to 2 units. The bounding surfaces (planes) are $AX + BY + CZ - D = 0$.

Surface #	A	B	C	D
1	1	0	0	0
2	1	0	0	2
3	0	1	0	0
4	0	1	0	2
5	0	0	1	0
6	0	0	1	2

The polarity of a point inside the volume element, say (1, 1, 1), must be (arbitrarily) positive. The surface list would then be 1, -2, 3, -4, 5, -6. Simply putting the coordinates of an internal point into each equation determines the polarity.

The input format for SIGMA is a standard FORTRAN scheme which may be obtained from the user manual.

4.2 MEVDP DATA FORMAT

The MEVDP code treats sparse geometry. Dense geometry is a subset of sparse geometry. Components are modeled in a two level scheme. The lower level consists of simple shaped volume elements, shown in Figure 3, called elemental volumes in the user manual. One to ten of these may be grouped to form a composite shield. At least one of the volume elements must contain material; some may be voids. Voids may embed solids within each composite shield (local embedding) but do not affect solids in other composite shields. From the coding, it appears that overlapping solids within a composite shield and between several shields are additive. It is not readily apparent whether such an arrangement would cause difficulties in the section which orders penetrations from outside to inside. Local embedding may be used effectively in MEVDP but overlap of solids should probably be avoided.

Unlike SIGMA, MEVDP does not require the user to develop surface coefficients. Instead, figure-related parameters are input. For a sphere, the coordinates of the center point and any surface point are input. For a hexahedron (box), the coordinates of the eight corner points are input. Similar parameters are required for other volume elements.

The use of figure-related parameters simplifies data preparation in some ways and complicates it in other ways.. The user is relieved of the burden of computing surface coefficients, filling surface tables, and arranging volume element tables listing surfaces. This approach permits the user to make certain blunders unless he is very careful. One example is that the eight corners of a hexahedron must be listed in a certain order. An incorrect order may lead to a nonsense volume element which diagnostics may not catch. Another example is that the points specifying a truncated ellipsoid must obey certain rules which are not clearly outlined in the user manual.

An explicit definition of the polarity of the MEVDP surfaces is not necessary. The points supplied for the convex volume elements implicitly assign polarity for internal use by the code.

The MEVDP input data format is listed on pages 94 - 97 of the user manual⁽¹⁹⁾. Coordinates and lengths must be in inches, angles in radians, and densities in gm/cm^2 . Conventional FORTRAN input is used so learning time is minimal. No provision is made for volume element identification other than by serial number. The user may wish to keep a separate tabulation of serial number descriptions, or put a title in columns 41 - 80 of the first card describing each composite shield.

4.3 HSV DATA FORMAT

The HSV code treats sparse geometry, as does MEVDP. The HSV code can treat more complex shapes than the simple geometric figures of MEVDP. Unlike the two level scheme of MEVDP - composite shields made up of simple volume elements - HSV uses a one level scheme based on volume elements. The HSV volume element may be rather complex in shape because up to 50 surfaces may bound it. It may be multiply connected or disjoint provided the polarity criterion is satisfied. Global embedding is permitted in HSV, whereas local embedding (within composite shields) is permitted in MEVDP.

For HSV the data pertaining to each volume element is input in arbitrary sequence. Required data include material number, density, an internal point, and any desired transformations. The density in gm/cm^2 is multiplied by 2.54 if distances are in inches rather than centimeters. The internal point is used by the code to set polarities of the surfaces. One or two transformations may be used to move the volume element.

Other input data include the number of plane surfaces and higher order surfaces. For HSV three points are specified for each surface in the volume element. Surfaces other than planes require two or three other parameters as specified in the user manual.

Data preparation for HSV is simplified by two special input features. These features are:

- preservation of input values, and
- use of NAMELIST input.

Preservation of input values means that, with a few exceptions, data

are read into an input section and remain unchanged until updated. NAMELIST is a format-free input-output routine suitable for FORTRAN programs. Descriptions of NAMELIST may be found in most FORTRAN user manuals for CDC, IBM, UNIVAC, and possibly others computers. The use of NAMELIST in combination with the preservation of input value feature is a powerful tool because it permits the user to update variables selectively. When modeling similar volume elements sequentially, the quantity of data for all but the first is often reduced by a factor of three to five.

In using HSV it is important to remember that the sequential location of volume elements is critical where embedding is used. For example, atmosphere may be added to a space vehicle by adding one or a few volume elements at the end of the deck. However, if the atmosphere volume elements are placed at the beginning of the deck, the embedding feature will erase the inboard equipment.

5.0 DATA CHECKING

Data checking is extremely important for complex geometry models. It is also very difficult. Liley and Hill estimate that at least half the effort in developing geometry models is devoted to data checking. Nevertheless, it is probable that large models still contain errors.

Four large models are known to exist. They are the Apollo Block II Command and Service Module with 968 volume elements, the Lunar Module⁽¹⁸⁾ with 1038 volume elements, a man model⁽¹⁷⁾ with 2518 and 2271 volume elements for the standing and seated version respectively, and the Skylab cluster with 3000 to 5000 volume elements depending on the configuration. Three of these models have been checked by organizations other than the originator and have been found to contain errors.

A realistic goal in constructing a large geometry model is to reduce errors to the point where the desired answers are not significantly affected. The elimination of all errors is usually too expensive to be justifiable.

The data checking procedures of the SIGMA, MEVDP, and HSV codes will be outlined. It is significant that each of these codes has a visual plot routine available. Visual plots offer the best checking assistance to the user.

5.1 SIGMA DATA CHECKING

Standard FORTRAN input routines are used so that some format errors are detected. These errors include decimal point in an integer field, comma instead of a decimal point, and other illegal characters. Certain card arrangement errors may also be detected.

Three error detecting schemes are incorporated into the SIGMA code. The first test deals with points in volume elements. Points are placed in each volume element and tested against all volume elements with the polarity criterion. Provision is also made for scattering test points at random through the model. If any point is contained in more than one volume element or in none, an error message is sent.

The second test is performed during ray tracing. Should the ray pass through an unspecified region, a "lost" signal is sent.

The third test concerns visual plots of cross sections through the configuration. The rectilinear plots are parallel to the coordinate planes. The scale factor is the same in both directions. Dots are used to show boundary crossings so that an outline sketch is produced. An example plot of a simple Skylab model is shown in Figure 13.

5.2 MEVDP DATA CHECKING

Standard FORTRAN input routines are used. As with SIGMA, certain illegal characters and card misarrangements may be detected.

Apparently no internal logic checks are incorporated into the MEVDP code to detect erroneous geometry data. Liley and Hamilton have written a separate program⁽¹⁸⁾ to scan and correct hexahedron data. The hexahedron is defined by the eight corner points. This method overdetermines each of the six planar faces by specifying four points in each plane. The auxiliary program checks the four points on each plane. If they do not define a perfect hexahedron, certain points are moved until this requirement is met. Any

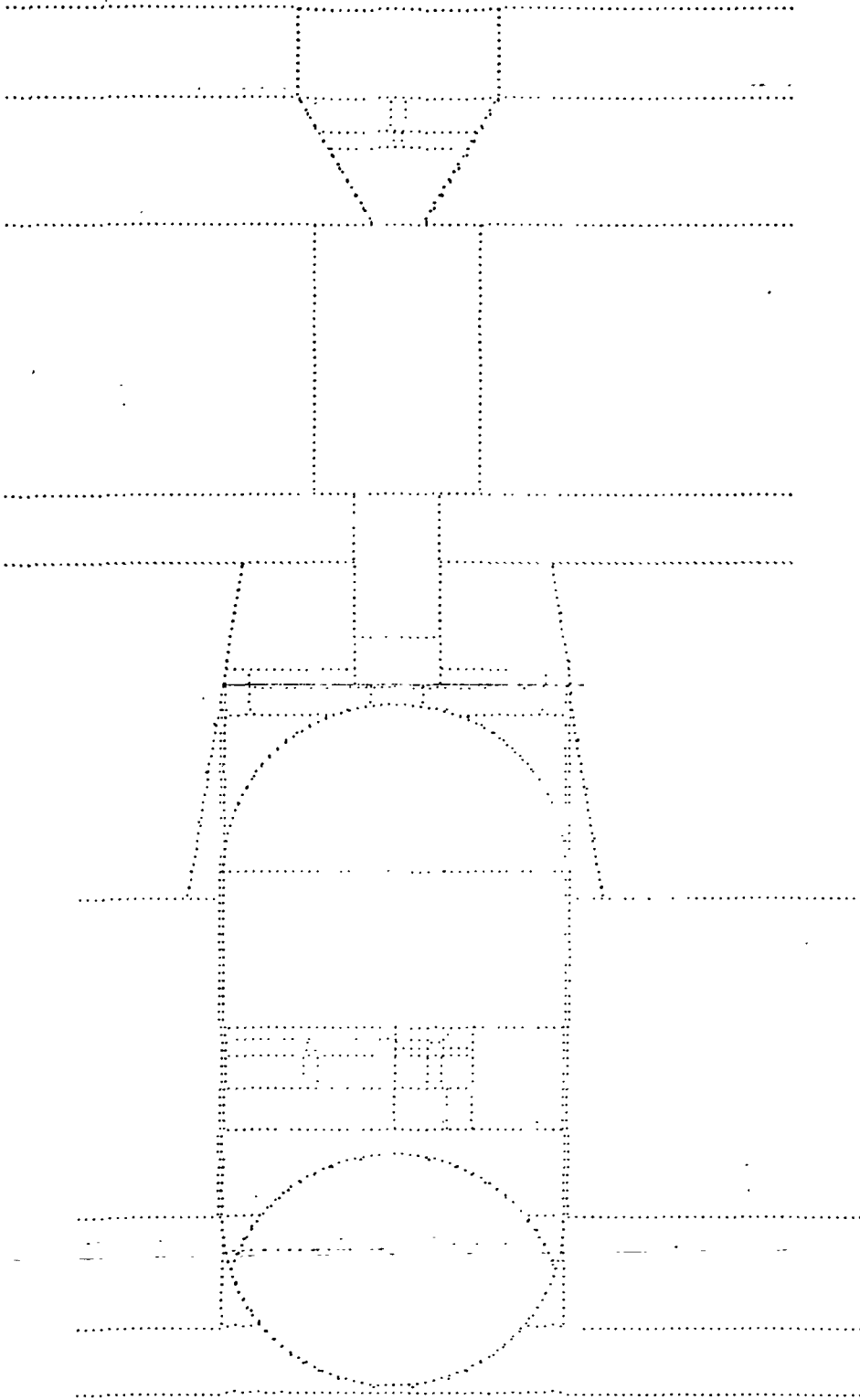


Figure 13. Sigma-Drawn Section Through Skylab Geometric Model²⁵

adjustment of more than ten inches is flagged for user inspection. The auxiliary hexahedron scanning program is not available to the general user. However, the method is outlined in a Lunar Module report⁽¹⁸⁾ and may be easily programmed.

Although MEVDP contains no internal logic checks, the voluminous output should be helpful in detecting errors. The output includes card images of input volume element data and specifications for the optional man models that may be called up. Next the dose point locations are given. Optionally, additional data are printed on the volume elements after they are transformed into the Rotated Detector Coordinate System.

The primary method of verifying MEVDP data uses visual plots from a CRT plotter. An example plot is shown in Figure 14. The user specifies the size and orientation of the cross section which need not be parallel to a coordinate plane. Rays are drawn radially outward from the center. Traces through matter are indicated as shown, one symbol per material. Overlaps of different materials may often be detected.

Liley points out that the radial plot simulates ray tracing from a dose point. The best resolution is nearest the center where errors may be more serious. It may be necessary to reduce gain on the CRT or to magnify the plot to avert blackout or washout near the center of the photographic image. An advantage of the radial plot over a rectilinear plot is that thin sections are less likely to be missed.

Each volume element should be verified by plots in several orientations. Small clusters of volume elements should be scanned

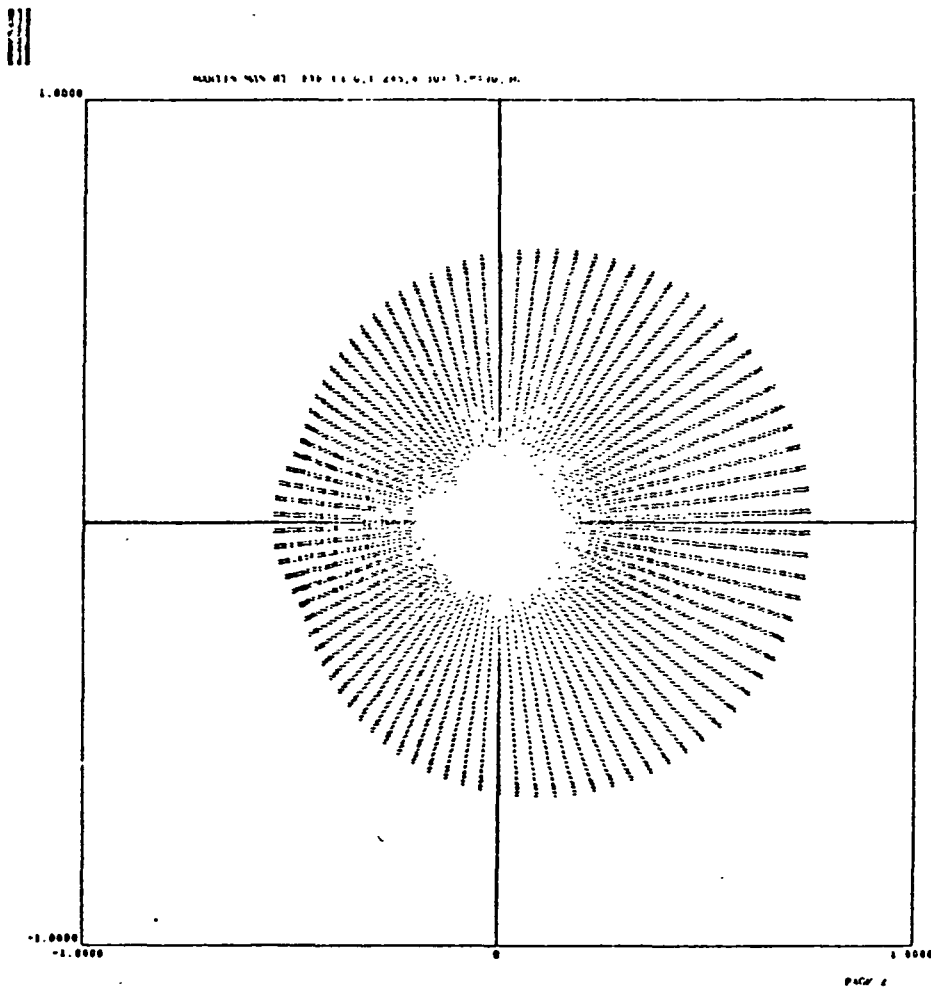


Figure 14. MEVDP Computer Drawing of the Eye¹⁷

to verify correct location and detect overlaps. The integration of small clusters into larger clusters and finally into the entire model should be checked at each step.

5.3 HSV DATA CHECKING

The HSV code uses the NAMELIST input routine which has its own diagnostics to detect illegal characters and improper sequences of data. Typographical errors seem to occur more frequently in NAMELIST data than in FORTRAN data. Also, present NAMELIST routines, like FORTRAN routines, stop processing data when an error is detected. Calendar time may be saved by subdividing a data deck and submitting each set with a short program that simply reads the cards and flags errors.

The HSV code has several schemes to detect logic errors in geometry data. Three types of errors are detected in all types of geometry data. First, the internal point used to set surface polarities must not lie on a surface. Second, the three points defining each surface must not be colinear. Third, two or more points defining each surface must not be identical. The third type of error is a subset of the second but code peculiarities usually cause an error comment pertaining to the first (internal point) type of error.

Several other logic checks are applied in those cases where the user speeds up ray tracing by providing sphere test parameters and super region data.

As described earlier, the sphere test will be performed during ray tracing if the user inputs a length to serve as the radius of

a sphere circumscribing the volume element. The internal point serves as the center of the sphere. The sphere parameter is automatically generated for converted MEVDP data.

If the sphere parameter is furnished, HSV will perform a logic check on the volume element immediately after the surface coefficients have been computed. HSV computes the coordinates of 50 points which lie along two great circles of the circumscribing sphere, but which are 0.1 length units outside the sphere. The great circles are at right angles to each other. The 50 points should all lie outside the volume element. HSV runs a fast polarity check to ensure that each point is on the negative side of at least one surface bounding the volume element. This test was originally put in to ensure that the user chose a sufficiently large radius for the sphere. However, it has proved to be a more general and useful test than anticipated. Thus far it has detected two cases in Skylab data where the sphere was too small and 20 cases in Skylab data where errors distorted the volume element.

As described earlier, the user may input pseudo volume elements called super regions around clusters of volume elements to permit fast scanning in the ray tracing part of HSV. The super region is started at the beginning of the cluster and turned off at the end of it. Super regions may be nested to a level 10 deep in a manner analogous to DO loops in a FORTRAN program.

During volume element testing, several logic checks are performed on super regions. First, the super region nesting must not exceed 10. Second, the super region count must never go negative, i. e., more ends than beginnings. Third, the super region count must go to zero at the end of the data. Fourth, the internal point of each

volume element must lie within all super regions which are supposed to contain it. Fifth, the internal point of each nested super region must lie within its nesting super regions.

The HSV format checks and logic checks described above detect approximately 80 to 90 percent of all geometry model errors. Most of the remaining errors may be detected by careful use of the visual plot routine. As with MEVDP, the recommended procedure is to verify each volume element in small groups. Additional plots should be made as groups of volume elements are integrated. It is important to take the plots in several orientations for each group.

The HSV plot routine is self-contained and puts the plots on the printer. An example plot is shown in Figure 15. Note that the HSV plotter fills in volume elements using a unique symbol for each one rather than for each material. This method allows differentiation between volume elements of like materials.

The HSV plots are rectilinear. The user specifies three corners of a rectangle (or trapezoid) for which he wishes a cross section plot. The orientation is arbitrary. Should the user specify three corners of a trapezoid (usually accidentally), it will be distorted into a rectangular plot. HSV sends a ray from the first point to the second and checks for volume element intersections. Ray segments penetrating non-void volume elements are filled in with a printer symbol. A parallel ray, displaced toward the third corner of the plot, is then checked. The process is repeated until the plot is finished. The user may specify different horizontal and vertical meshes (stretch) or the code will automatically compute equal mesh spacings.

Two tables follow each plot. The first table contains a list of each symbol used in the plot, the volume element number corresponding to each symbol, material number, and density. The second table was added because of the limited resolution of the plot which is 126 characters horizontally. Each line of the plot is numbered sequentially though not shown in Figure 15. The second table shows actual penetration lengths through volume elements for each line of the plot. In this way the penetration lengths may be known to five significant figures though the pictorial lengths have much poorer resolution.

HSV offers one additional check which is applied after routine checkout is finished and production runs start. For each ray processed, the code prints the dose point coordinates, direction cosines of the ray, solid angle represented by the ray, and a list of all volume element penetration thicknesses. Errors in placement of the dose point, outer wall thicknesses, and the mating of modules may be determined from this listing.

6.0 CODE COMPARISON AND EVALUATION

The evaluation of the three complex geometry codes discussed in this report is difficult. In many cases, a user shopping for a space radiation code to apply to his problem may bypass the geometry capabilities of each code entirely and may concern himself only with the treatment of dose calculations. Will a code treat protons, alphas, electrons, and bremsstrahlung? Will a code compute rad doses, rem doses, or photographic film fogging? Will a code produce the spectral flux hitting a dose point so that a new response function can be inserted? These are often the critical questions for a user and they have little to do with geometry. Nevertheless, it should be remembered that geometry considerations are often crucial in space radiation problems. Careful geometry modeling can reduce some of the uncertainties in a calculation enormously, perhaps from a factor of four down to 10 percent. The analyst is then free to concentrate on other matters such as source terms and radiation effects.

6.1 SIGMA GEOMETRY CAPABILITY

The SIGMA code is a logical derivative of earlier radiation transport codes. The user who has experience with point kernel and Monte Carlo geometry procedures should feel comfortable with SIGMA. Jordon has modified the SIGMA geometry routine slightly and put it in the KAP⁶(point kernel) and FASTER¹⁴ (Monte Carlo) codes which treat neutron and gamma ray transport, as well as BETA,¹⁵ an electron Monte Carlo code.

The similarity of SIGMA to many reactor geometry codes is a strong point in its favor. Langley and Billings also point out that

the same geometry model may be used in space radiation, reactor, and radioisotope calculations. A small geometry model may be set up quickly - typically a week - and verified to the point where a user has high confidence that no errors remain undetected. Further, a relatively simple shield optimization code is coded in SWORD which uses the SIGMA geometry.

SIGMA is alone among the three codes discussed in this report in being able to treat the general quadric. Special equation forms are available for surfaces parallel to coordinate axes.

The principal limitations of SIGMA are: 1) it is restricted to dense geometry, 2) a limited number of surfaces may bound a volume element, 3) configuration changes are difficult to accomodate, and 4) it is limited to relatively simple configurations. A new version of the code may soon remove limitations 2) and 4).

The dense geometry limitation is usually not serious in reactor problems. In space vehicles, however, inboard components which are difficult to model may reduce dose rate by a factor of two to ten or more. The user should be sure that the important components of his configuration are amenable to dense geometry treatment before he chooses SIGMA.

SIGMA has been restricted to having ten or fewer surfaces bounding a volume element. This limitation often requires the user to subdivide the geometry and use additional volume elements. Alternatively, the user may modify the code to accept additional surfaces per volume element. The yet unpublished SIGMA version has been so modified.

Configuration changes often are troublesome because of the interlocking way the geometry is tied together through the surface table and the volume element surface list. The accomodation of a change may require a large user effort.

Finally, the SIGMA code is usually restricted to 100 surfaces and 100 volume elements. This quantity of data is adequate for simple and moderately complex models, but it will not handle very complex models. The yet unpublished SIGMA version can analyze larger models.

6.2 MEVDP AND HSV GEOMETRY CAPABILITY

MEVDP and HSV are quite different from SIGMA and most other geometry codes, but are similar to each other in many respects. Both are designed to treat sparse geometry. Both will analyze simple and complex models in the one to 5000 volume element range with facility. Both could be extended to treat at least 10,000 volume elements with minor modifications.

The disadvantages of MEVDP and HSV include the following items.

- 1) Longer computer time than SIGMA.
- 2) Difficulty in detecting all errors.
- 3) Redundant input.

The speed of sparse geometry codes should usually be lower than the speed of dense geometry codes. The disparity is due to the priority scan tables which may be set up automatically in dense geometries. Relative speeds will be discussed below.

Error detection schemes in geometry codes need further development. At present it is relatively difficult to find errors in large

geometry models. Diligent application of available techniques will usually reduce errors to the point where results are not materially affected. In error detection capability, SIGMA is probably the best, HSV is next, and MEVDP is last.

Both MEVDP and HSV require redundant data. A particular surface may bound several - or even several hundred - volume elements. Yet the surface must be input for each of those volume elements. The reason for the redundancy is as follows. For MEVDP the user inputs body-oriented data rather than surface-oriented data. A computer search for identical surfaces would be too time consuming. For HSV the user inputs surface-oriented data but the task of recording and tracking thousands of surfaces would be too laborious. Further, the extra surface data does not require a larger core because surface data is placed in mass storage.

Despite the similarities between MEVDP and HSV, major differences exist as well. MEVDP is a body-oriented system with a two level volume element and composite shield organization. Local embedding is enforced within composite shields. This feature is required because of the limited shapes permitted. HSV is more surface-oriented with a one volume element organization. Global embedding is enforced among all volume elements. The HSV volume elements may have a more complex shape than MEVDP volume elements.

The MEVDP input scheme is relatively easy to learn. The format is rigid which makes the rules for data preparation straightforward. HSV uses NAMELIST input which is more difficult to learn. Experience with both codes has shown that HSV data preparation is

harder than MEVDP data preparation initially, but becomes easier and faster after several weeks experience.

In a previous section it was shown that more flexible codes generally require fewer volume elements. In order to model a simple insulated can containing a liquid, SIGMA requires seven volume elements; MEVDP, five, and HSV, three.

MEVDP uses several techniques to increase ray tracing speed. These techniques are completely built in; no user assistance is required. HSV also uses several techniques to increase ray tracing speed, but user assistance is required.

A change in the location of volume elements is relatively difficult in MEVDP, but is fairly simple in HSV through the available transformations.

MEVDP has the capability of combining diverse materials into an equivalent thickness of one material. The code computes cumulative solid angle versus shield thickness and fits functions to this table. The functional fits may be used for dose calculations with diverse spectra without repeated ray tracing.

HSV stores ray tracing data on tape or mass storage. These data may be used for dose calculations with diverse spectra without repeated ray tracing. The penetration thickness data is tagged by material numbers so that the density of selected components may be varied without further ray tracing. The latter feature is helpful in design optimization studies, and removal of components.

6.3 CODE SPEED COMPARISONS

SIGMA is certainly the fastest of the three codes considered. Its speed is made possible by dense geometry, surface-volume element

correlation, a priority (guess) list for next volume element, and classification of surfaces into 4, 7, or 10 coefficient categories.

Estimates of relative speed are approximate because the speed of each code may vary by a factor of three depending on the location of the dose point. Also, for HSV, the volume element setup penalty on the Univac 1108 is 60 seconds per run for 3000 volume elements. The setup penalty can change the speed of HSV by a factor of two depending on the number of rays traced. For MEVDP, the transformation penalty on the CDC 6500 for each dose point is 100 seconds for a 2500 volume element model. This penalty may be relatively large or small depending on the number of rays per dose point. Another factor affecting speed is the flexibility of the code. The number of volume elements required to model a configuration may vary by a factor of two or more between codes.

For the reasons given above, comparisons of operating speed are always suspect. This caution should be kept in mind in analyzing the relative speeds shown in Table 6.1. Cases 1 - 4 are SIGMA runs made by Billings. Cases 5 - 6 are MEVDP runs made by Billings. Cases 7 - 12 are MEVDP runs made by Kase.¹⁷ Cases 13 - 15 are EVDP runs made by Liley. Cases 16 - 19 are HSV runs made by Hill. Case 6 is identical to Case 5 except that the sorting of penetrations into sequence is omitted. Two measures of speed are shown. One is rays per second. Sigma runs from 6.7 to 22 rays per second in these cases. MEVDP runs from 0.42 to 0.67 rays per second. EVDP runs from 1.4 to 2.0 rays per second. HSV runs from 0.72 to 1.5 rays per second.

Table 6.1 Relative Speed of Geometry Codes

Case	Code	Geometry Model	Computer	No. Rays	Time (Sec)	Rays/Sec.*	E* Figure of Merit	Source	Notes
1	SIGMA	Man Model 1200 Surf 2550 VE	CDC 6500	400	60	6.70	17000	Billings	1, 2
2	SIGMA	Man Model 44 Surf 44 VE	CDC 6500	500	23	22.00	957	Billings	3
3	SIGMA	Part Man Model 326 Surf 699 VE	CDC 6500	400	25	16.00	11184	Billings	1
4	SIGMA	Part Man Model 326 Surf 699 VE	Univac 1108	400	14	16.00	11184	Billings	1, 4
5	MEVDP	Man Model 557 CS 2518 VE	CDC 6500	512	1100	0.47	1164	Billings	5
6	MEVDP	Man Model 557 CS 2518 VE	CDC 6500	512	770	0.67	1662	Billings	5, 6
7	MEVDP	Man Model 504 CS 2271 VE	CDC 6600	512	421	0.61	1381	Kase	7, 8
8	MEVDP	Man Model 504 CS 2271 VE	CDC 6600	512	446	0.57	1304	Kase	7, 8
9	MEVDP	Man Model 504 CS 2271 VE	CDC 6600	512	502	0.51	1158	Kase	7, 8
10	MEVDP	Man Model 557 CS 2518 VE	CDC 6600	512	431	0.59	1496	Kase	5, 8
11	MEVDP	Man Model 504 CS 2271 VE	CDC 6600	200	212	0.47	1071	Kase	7, 8
12	MEVDP	Man Model 557 CS 2518 VE	CDC 6600	512	616	0.42	1047	Kase	5, 8
13	EVDP	LM 1038 VE	IBM 360/55	512	638	1.50	1522	Liley	9, 10, 11, 12
14	EVDP	CSM Tunnel 1171 VE	IBM 7094	512	656	2.00	2377	Liley	9, 12, 13, 14
15	LVDP	CSM Belt Buckle 1171 VE	IBM 7094	512	922	1.40	1691	Liley	9, 12, 13, 14
16	HSV	Skylab 3229 VE	Univac 1108	660	316	1.50	4817	Hill	15, 16
17	HSV	Skylab 3309 VE	Univac 1108	264	164	1.10	3805	Hill	15, 16
18	HSV	Skylab 3355 VE	Univac 1108	108	107	0.72	2419	Hill	15, 16
19	HSV	Skylab 3355 VE	Univac 1108	276	181	1.10	3654	Hill	15, 16

*Normalized to CDC 6500.

NOTES TO FIGURE 6.1

1. The standing man model devised by Kase in the MEVDP system was manually converted to the SIGMA system by Billings. Code modifications were added to SIGMA to correct for remaining overlays and gaps. Cases 1 and 5 are comparable.
2. The SIGMA man model contains 1200 surfaces (surf) and 2550 volume elements (VE).
3. A simple man model and simple vehicle model are included.
4. Cases 3 and 4 are identical except for the computer used.
5. The Kase standing man model contains 2518 volume elements arranged into 557 composite shields (CS).
6. Case 6 is identical to case 5 except that material penetrations are not sorted and arranged in order. Sorting requires 330 seconds. Sort time is always included for SIGMA and HSV.
7. The Kase seated man model contains 2271 volume elements arranged into 504 composite shields.
8. The CDC 6600 is two to four times faster than the CDC 6500. An optimistic factor of two is chosen for normalization purposes. A larger factor would reduce the efficiency factors.
9. The EVDP code is an earlier version of MEVDP. Liley states that ray tracing times are comparable.
10. LM refers to the Lunar Module geometry model.
11. The CDC 6500 is assumed to be twice as fast as the IBM 360/55.
12. Time includes transformation from vehicle to dose point coordinate system (42 to 61 seconds) and fitting thickness versus solid angle functions (2 to 6 seconds), as well as ray tracing time. Dose and spectrum calculation times are omitted. Material penetration sorting times are not included.
13. The CDC 6500 is assumed to be 2.6 times faster than the IBM 7094.
14. CSM refers to the Apollo Command and Service Module. Tunnel and Belt Buckle refer to dose point locations.
15. Dose points are located at several places in the Skylab cluster including the Wardroom, Multiple Docking Adapter, and Film Vault.
16. The CDC 6500 and CDC 6400 have the same speed. The Univac 1108 is 1.4 times faster than the CDC 6400 for HSV.

Another measure of speed is the figure of merit, E, defined as follows:

$$E = \frac{RxV}{TxK}$$

where R is the number of rays, V is the number of volume elements, T is the time in seconds, and K is a speed normalization factor between computers.

The reason for choosing this form for E is that MEVDP may check every volume element for every ray. Therefore time should be approximately proportional to the product of rays times volume elements. HSV would do the same except that super regions permit skipping over most of the volume elements. SIGMA times should be proportional to the number of rays times volume elements penetrated.

The range of E for SIGMA is 957 to 17000 for the cases shown. The range of E for MEVDP is 1164 to 1662 for the cases shown. The range of E for EVDP is 1522 to 2377 for the cases shown. The range of E for HSV is 2419 to 4817 for the cases shown. Note that E is not expected to be constant for SIGMA and does, indeed, show a wide variation.

The most valid comparison in Table 6.1 is between cases 1 and 5 where, for nearly identical models, SIGMA is 14 times faster than MEVDP.

Cases 3 and 4 are identical except for machines. These cases show that the Univac 1108 is 1.8 times faster than the CDC 6500 for small models with SIGMA. Similar cases (not shown) yield a consistent factor of 1.4 for large models with HSV.

Similar comparisons for these codes on the CDC 6500 and CDC 6600 are not available. Experience with other codes shows the CDC 6600 to be faster by a factor of two to four. An optimistic factor of two is chosen to normalize the speed of MEVDP in cases 7 - 12.

Again, speed comparisons are somewhat uncertain due to model differences, modeling capabilities, setup times, machine normalization, and the unfairness of choosing either rays per second or the figure of merit as defined above as a measure. With these cautions in mind, it appears that, for the cases shown, HSV is two to four times faster than MEVDP, and SIGMA is three to five times faster than HSV.

6.4 EVALUATION SUMMARY

In summary, SIGMA offers large advantages in speed and learning time for models where dense geometry is feasible. Error checking is moderately good. When the new SIGMA version becomes available, large models may be treated. Insufficient experience is available to show whether the preparation of tables containing thousands of surface and volume element cross references is easy or difficult. Configuration changes are difficult in SIGMA.

MEVDP offers advantages in learning time and sparse geometry. A new version not yet documented adds elliptical cones and cylinders to its capabilities. The new version also treats batches of HSV data intermixed with MEVDP data. The conversion is usually one MEVDP volume element per HSV volume element, but may go as high as eight to one for odd shapes. Some HSV volume elements must be subdivided before conversion. The speed of MEVDP is low. Additional error checking techniques would be desirable. The potential user should be sure that the CRT plot routines will work at his installation. Configuration changes may be made by hand or the user may write his own transformation program.

HSV offers advantages in speed of data preparation and sparse geometry. The quantity of data per volume element is large but the data preservation feature greatly reduces the effort. The flexibility of HSV means that a model requires fewer volume elements than the other codes. An option is available to convert MEVDP volume elements to HSV volume elements on a one-to-one basis. The MEVDP data may be intermixed with HSV data.

The speed of HSV is fairly high if the user prepares super regions. Error checking is moderately good. Configuration changes are easy to handle with the double transformation. HSV requires more learning time than the other codes because of NAMELIST input and general volume element shapes.

Data preparation time for the three codes varies with the code, the user, and model complexity. Most individuals can prepare and check model data at the rate of one half to three volume elements per man hour. Subjective comments on data preparation for each code follow.

SIGMA - Data preparation for SIGMA appears to be designed for meticulous, painstaking individuals who like to prepare large interlocking tables into an integrated unit before processing.

MEVDP - Data preparation for MEVDP follows rigid rules for generating data for simple shapes. A large model has been generated by high school graduates with minimal training. Data may be checked in small batches prior to integration into the model.

HSV - Data preparation for HSV is designed for technical personnel who are willing to learn a flexible input system in order to minimize tedious data transcription. Data may be checked in small batches prior to integration into the model.

7.0 RECOMMENDATIONS FOR FUTURE GEOMETRY DEVELOPMENTS

Should a new space radiation complex geometry shielding code be written? The answer to this question is: "Not Yet". The reason for the above answer is that the three codes discussed in this report can treat present space radiation problems with reasonable competence and efficiency.

The development of new geometry codes will be triggered by new types of problems beyond the capability of present codes. Such new types of problems are rarely forecast in advance. In the space radiation area, new capabilities might eventually be required in treating electrons, secondaries, and galactic cosmic radiations.

However, the present codes are certainly not perfect. Each code has some growth potential and near term efforts should be devoted to improving these codes. The discussion of suggested improvements will include geometry capabilities, speed, and usability.

7.1 SUGGESTED SIGMA IMPROVEMENTS

Most of the suggested improvements for SIGMA are tardy. Recent unpublished modifications include many of the planned suggestions. One suggestion is to enlarge the capacity of SIGMA from 100 surfaces and 100 volume elements. Another is to remove the restriction on the number of surfaces per volume element and to pack the surface list to avoid the unused words.

Other suggestions pertain to ways of speeding up the code. The old SIGMA version used a priority list to guess the next volume element after leaving one. The code could update the list as it traces its way through the model. The list should be two deep at a minimum.

At the expense of core storage, the list could be set up for each surface bounding the volume elements.

The new SIGMA version speeds the ray scan by cross correlating surfaces and volume elements. The user specifies each surface bounding a volume element. The code then inverts the procedure and lists all volume elements touching each surface.

After crossing a boundry, the code first checks the priority list to guess the volume element entered. If that test fails - possibly because the list is short - the code tries the cross reference list which should almost always contain the proper volume element. If the second test fails, the ray is either "lost" due to a gap in the model, or the ray has crossed a vertex where several volumes meet, or the user has input the same surface more than once. A detailed check on all volume elements should be made before the "lost" error is signaled.

The new version of SIGMA is fast enough that further speed increases are unnecessary for most applications. The cost of running the code is usually less than the cost of constructing the geometry model.

The task of setting up geometry data for SIGMA is easy for small models, but becomes much more difficult with larger models. It is not clear how the bookkeeping task might be simplified for the user, but some thought should be given to it.

At present, and in the new version, SIGMA possesses one severe limitation which could be partially removed. The dense geometry feature makes it impossible to model a spacecraft adequately because the thousands of items of inboard equipment cannot be taken into account. In one case, the simple geometry model of Skylab devised for SIGMA

predicts an effective shield of 2 gm/cm^2 in the Workshop. HSV using a more detailed model predicts an effective shield of 10 gm/cm^2 or more. Thus inboard equipment is often important in transport calculations.

It is possible to convert SIGMA into a hybrid dense-sparse geometry code simply by permitting embedding and adding new logic paths to the code. Second level volume elements could be embedded completely within flagged volume elements. If a ray enters a flagged volume element, a check would be made to determine penetration of embedded volume elements. Several levels of embedding could be permitted. The dense geometry restriction could be enforced among groups of embedded volume elements to speed ray tracing.

Several advantages would accrue with the embedding feature. Models of complex configurations such as the Skylab and other spacecraft could be analyzed by SIGMA. SIGMA cannot presently handle such models. The number of volume elements in a model could be reduced. SIGMA requires six volume elements to treat a hollow box. Embedding would reduce this to two as is done with MEVDP and HSV. Running times for present (dense) models would not be affected. The code would slow down when embedded elements were encountered but would run nearly as fast otherwise.

The prime difficulty in adding embedding to SIGMA would be in devising a scheme of flagging and indexing which is simple enough to use in constructing geometry models. It appears to be impossible to match the simple local embedding of MEVDP or the global embedding of HSV.

Error checks performed by SIGMA could be improved. At present, the internal points and random points are checked to ensure that

they are within one and only one volume element. The random points may also lie on a boundary. A ray tracing scheme to "weigh" each volume element may be included in the new code. Finally, printer plots may be produced to provide an outline cross section through the model.

The outline plots are helpful, Figure 13, but filled-in plots are much better for complex models, Figures 14, 15. A unique symbol may be assigned to each material as in MEVDP or, better, to each volume element as in HSV. Void volume elements may be left blank. A cross reference between symbols and material or volume element should be given. The coarse resolution of the plots may cause some adjacent thin sections to be omitted. Even when plotted, the thickness is often hard to judge. A listing of symbols and penetration thickness for each ray of the plot is a great help.

7.2 SUGGESTED MEVDP IMPROVEMENTS

The present MEVDP code does not treat elliptical cones and cylinders. A new version which may be available shortly has remedied this lack. Kase¹⁷ has recommended the above changes and one other not yet implemented. He found a need for voiding the exterior of a volume element within a composite shield.

A transformation option similar to that of HSV would simplify data preparation and the accomodation of configuration changes. MEVDP now allows the user to bring in a particular group of volume elements (simple man model) up to 10 times and transform it to the desired location. This feature should be extended to other groups such as fuel tanks, batteries, etc., to simplify data preparation.

Of the right points used to specify a hexahedron, two are redundant. Thus, the hexahedron of Figure 16a may be completely defined by points 1, 2, 3, 5, 6 and 7. With six points the problem of

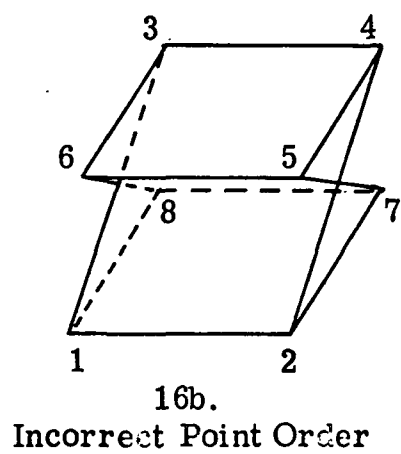
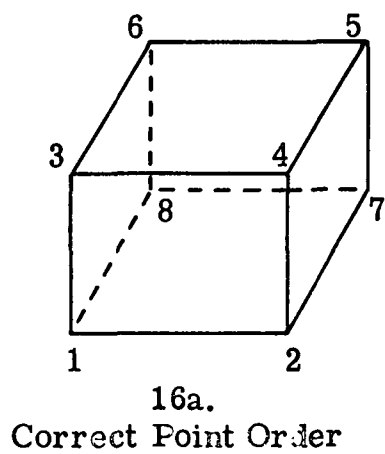


Figure 16. MEVDP Hexahedron Point Ordering.

ensuring that planes are unambiguous vanishes. If the eight point requirement is kept, the redundant information should be used for error checking. If eight points are used, out-of-order points may cause opposite faces of the hexahedron to intersect as shown in Figure 16b. This error was detected in six volume elements of the Apollo Block II Command and Service Module geometry model after it had been in use several years. It was detected then only because the model was converted to HSV format and HSV error diagnostics caught it.

Another test which might be added to the MEVDP code concerns the ellipsoid. The four basic points defining the ellipsoid must constitute a right-handed, orthogonal system. Simple vector calculations may be coded to check both orthogonality and handedness.

The tests suggested above require no user assistance. Two other tests could be added with user assistance. First, a flag could be added to rectangular hexahedra to signal the code to check for orthogonality. Second, thin uniform thickness hexahedra could be flagged to show which faces are intended to be parallel. The MEVDP code could send several vectors through these faces to check for uniform thickness.

As indicated in earlier sections, MEVDP is the slowest of the three present codes despite several fast scan tests. MEVDP must check every volume element (except some voids) for every ray. This code could be much faster if the HSV super region option were provided to the user.

The MEVDP CRT plot routine is one of the few error checks available in this code. It works well in those installations which have compatible hardware and software. Problems may occur when equipment

is changed or the code is used at a new installation. For example, Kase modeled 3156 volume elements for the model man in MEVDP format. Because the CRT plot routine did not work at his location, he limited his composite shields to one solid and up to eight void volume elements each. He converted the MEVDP data to HSV (then LSVDC4) format by hand and used the HSV plot routine for initial checking. This method gives an inadequate check because the input data is different. Today, of course, he could use the MEVDP to HSV conversion routine and perform a better check.

The point is that codes should be as machine-independent as is possible. The MEVDP CRT plot routine should be replaced by a printer plot routine. The symbols should be unique to volume elements rather than material. A list of symbols versus volume element, material, and density should be provided as is done by HSV. A list of penetration thickness by ray would increase plot resolution in one direction as is done in HSV. These plot routine changes should enhance error checking.

7.3 SUGGESTED HSV IMPROVEMENTS

The setup time for 3300 volume elements is about 60 seconds on the Univac 1108. This time is spent in reading in the data, computing all surface coefficients, and performing certain error checks. The setup time is incurred once per run regardless of the number of dose points run. These coefficients should be saved for future runs, thus reducing most setup times from 60 seconds to a few seconds. The complete setup should be performed only when the geometry model is changed.

One of the difficulties of treating large geometry models is the sheer bulk of data. The data are usually put on cards first,

then transferred to tape or mass storage and updated as needed. HSV is designed to minimize the card images required, but one additional option could be incorporated to reduce data transcription and card punching dramatically - perhaps to less than one card per volume element on the average. The savings is made possible by the normal repetition of components in most configurations. Identical trusses, longerons, fuel tanks, film magazines, etc., are currently repeated in the card deck. A few simple logic changes would make it possible to input such repeated clusters of volume elements into a library file. Then one or two cards in the data stream would suffice to call up a cluster of volume elements and transform it to the proper location.

Two examples of possible uses will be given. First, Kase¹⁷ devised 3156 volume elements in the MEVDP system for his model man. A total of 13,743 cards were punched. The model man does not possess as much redundancy as most space vehicles, but many components occur in pairs. The right limbs comprising 604 volume elements could be placed in the library. Two cards in the data stream could call them out. Then another two cards could call them out again, apply an improper rotation to make them left limbs, and move them to their proper location. The same method could be applied to the eyes, ears, and other symmetrical organs. It is estimated that at least 30 percent or 4200 cards could be saved.

The second example is the HSV Skylab model. Here, 3322 volume elements are placed on 10,173 cards. One third of the cards contain only identifying comments helpful to the user but ignored by the program. Thus HSV requires, on the average, two data cards plus an

optional comment card per volume element compared to 4.3 data cards per volume element for MEVDP. The effective ratio between codes is even larger because the man model would have required less than half as many volume elements if it had been modeled for HSV rather than MEVDP. At any rate, the Skylab cluster could be reduced from 10,173 cards to less than 3000 cards, including comments, if the library option were available. The ratio would then be about one card per volume element.

The logic structure of HSV permits the super region feature to be retained if the library option is added. Indeed, it would be easier to add more super regions and increase speed appreciably with the library option.

Several modifications could be made to the super region option of HSV to speed data preparation and checkout. First the super region start and end cards could be labeled to help ensure a correct relative positioning in the data stream. Second, a general volume element type super region would supplement the present cylindrical super region. Third, the setup of simple shape super regions could be automated further so that the generation of surface data would be performed by the code from a few points and lengths. Fourth, the sphere test parameter could be generated internally for simple super region shapes.

Finally, a weighing option using ray tracing techniques appears desirable for most users.

REFERENCES

1. Allen, R. I.; Bly, F. T.; Dessler, A. J.; Douglass, C. C.; Perkins, J. F.; Price, H. C.; Schofield, W. M.; Smith, E. C.; Editor Tolan, J. H.: Shielding Problems in Manned Space Vehicles-Annual Report-1960, NR-140, Lockheed-Georgia Company, September 1960.
2. Alsmiller Jr., R. G.; Sanatoro, R. T.; Barish, J.; and Claiborne, H. C.: Shielding of Manned Space Vehicles Against Protons and Alpha Particles; ORNL-RSIC-35, Oak Ridge National Laboratory, November 1972.
3. Billings, M. P. and Langley, R. W.: A Technique for Evaluation of Space Radiation Dose to Distributed Body Organs, Presented to the ANS Annual Meeting, Boston, Mass., 13-17 June 1971, MDAC WD1630, October 1971.
4. Bresticker, S.: Grumman Aircraft, Private Communication, 1968.
5. Burrell, M. O.: The Calculation of Proton Penetration and Dose Rates, Second Symposium on Protection Against Radiation In Space - October 1964, NASA SP-71, 1965.
6. Disney, R. K. and Zeigler, S. L.: Nuclear Rocket Shielding Methods, Modification, Updating, an Input Data Preparation, Volume 6, Point Kernel Techniques, WANL-PR-(LL)-034, Westinghouse Astronuclear August 1970.
7. Fortney, R. E.: Computer Analysis of Radiation Shielding, AMRL-TDR-64-11, (NSL 63-168), Northrop Space Laboratories, February 1964.
8. Hill, C. W.; Ritchie, W. B.; Simpson, K. M.: Data Compilation and Evaluation of Space Shielding Problems - Dose Calculations In Space Vehicles, Volume II, ER 7777, Lockheed-Georgia Co., Marietta, Ga.; August, 1965.

9. Hill, C. W.; Ritchie, W.B.; K.M., Jr.: Data Compilation and Evaluation of Space Shielding Problems - Dose Calculations In Space Vehicles, Volume II, ER 7777, Lockheed-Georgia, Co., Marietta, Ga.; August 1965.
10. Hill, C. W.; Ritchie, W.B.; and Simpson, K.M., Jr: Data Compilation and Evaluation of Space Shielding Problems Volume IV, LSVDC4 Program Sustem, ER-7777, Lockheed-Georgia Company, Marietta, Georgia, 1967.
11. Hill, C. W. and Neville, C. F.: Space Radiation Hazards to Project Skylab Photographic Film-Phase II, NASA CR-61364 (ER-11192), Lockheed-Georgia Company, October 1971.
12. Hill, C. W.: HSV User Manual, (to be published).
13. Jordan, T.M.: SIGMA, A Computer Program for Space Radiation Dose Analysis Within Complex Configuration, DAC-60878, Douglas Aircraft Company, November 1967.
14. Jordan, T.M.: Synthesis of Calculational Methods for the Design and Analysis of Radiation Shields for Nuclear Rocket Systems, WANL-PR-(LL)-014, Westinghouse Astronuclear, December 1967.
15. Jordan, T.M.; BETA, A Monte Carlo Computer Program for Bremsstrahlung and Electron Transport Analysis, AFWL-TR-68-111, A.R.T. Research Corporation, October 1968.
16. Kahn, H.: Applications of Monte Carlo, RM-1237-AEC, The Rand Corporation, April 1954 (Revised April 1956).
17. Kase, P.G.: Computerized Anatomical Model Man, Air Force Weapons Laboratory Report AFWL-TR-69-161, Martin Marietta, January 1970.
18. Liley, B.; Hamilton, S.C.; and Fletcher, J.D.: Numerical Model Representing the Geometrical Shielding of the Lunar Module, SD68-798, North American Rockwell, September 1968.
19. Liley, B. and Hamilton, S.C.: Modified Elemental Volume Dose Program (MEVDP) AFWL-TR-69-68, North American Rockwell, March 1971.

20. Madey, R.; Levine, D.; Schwamb, F.E.; and Sisavic, F.M.: A General Geometry Shielding Code For Space Protons, Trans. American Nuclear Society, 1, Number 1, June 1964.
21. Malone, C.F.: Program K019, Shield Thickness Calculations Program - Volume I, Computer Users Manual, NASA - MSC-3066, May 1965.
22. Oden, L.: Metoder Och Rutiner For Berakning I Datamaskin Av Skarning Mellan En Rat Linje Och Rymdgeometrisk Figurer (Computer Methods & Routines for Calculations Straight Line Intersections with Solid Geometry Figures) (In Swedish), FOA-2C-2212-45(N69-35475), December 1967.
23. Shreve, D. C. and Lonergan, J.A.: Electron Transport and Space Shielding Handbook, SAI 71-559-LJ, Science Applications, Inc., November 1971.
24. Simpson, K.M.; Hill, C. W.; and Douglass, C.C.: A Space Radiation Shielding Code for Realistic Vehicle Geometries, Second Symposium on Protection Against Radiations in Space - October 1964, NASA SP-71, 1965.
25. Yucker, W.R. and Lilley, J.R.: Charge Code for Space Radiation Shielding Analysis, DAC-62231, McDonnell Douglass, April 1969.